One Identity Safeguard for Privileged Sessions 6.9 Security Target

Version 1.0
January 20, 2022

Prepared for:

One Identity LLC
4 Polaris Way
Aliso Viejo, CA 92656
United States

Prepared by:

leidos

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

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.

The Security Target contains the following additional sections:

- Product and TOE Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

1.1 Security Target, TOE and CC Identification

ST Title – One Identity Safeguard for Privileged Sessions 6.9 Security Target

ST Version – 1.0

ST Date – January 20, 2022

TOE Identification – One Identity Safeguard for Privileged Sessions v6.9.4

The TOE consists of the following hardware components:

- Model 3000 with Intel Xeon E3-1275 CPU (Kaby Lake)
- Model 3500 with 2 x Intel Xeon Silver 4110 CPU (Skylake)

These hardware components include the One Identity Safeguard for Privileged Sessions v6.9 software/firmware, which runs on a modified version of Ubuntu Linux LTS 18.04 as its base operating system plus additional software/firmware needed to support the product's functionality. The software/firmware is obtained and installed as a single bundle that includes the underlying operating system; it is not installed as a separate application on a general-purpose server operating system.

1.2 Conformance Claims

This TOE is conformant to the following CC specifications:

- collaborative Protection Profile for Network Devices, Version 2.2e, 23-March-2020, NDcPP and including the following optional and selection-based SFRs: FCS_HTTPS_EXT.1, FCS_SSHC_EXT.1, FCS_SSHS_EXT.1, 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, and FMT_MTD.1/Cryptokeys. The following NIAP Technical Decisions apply to this PP and have been accounted for in the ST development and the conduct of the evaluation:
  - 0592 – NIT Technical Decision for Local Storage of Audit Records
  - 0591 – NIT Technical Decision for Virtual TOEs and hypervisors
    - N/A to the TOE; the TOE does not have a virtual component or model
  - 0581 – NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3
    - N/A to the TOE; the TD adds a new selection option for FCS_CKM.2 but the TOE does not claim it
  - 0580 – NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e

- 0572 – NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers
- 0571 – NiT Technical Decision for Guidance on how to handle FIA_AFL.1
- 0570 – NiT Technical Decision for Clarification about FIA_AFL.1
- 0569 – NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7
  - N/A to the TOE; the TOE does not claim session resumption functionality for its TLS server
- 0564 – NiT Technical Decision for Vulnerability Analysis Search Criteria
- 0563 – NiT Technical Decision for Clarification of audit date information
- 0556 – NIT Technical Decision for RFC 5077 question
- 0555 – NIT Technical Decision for RFC Reference incorrect in TLSS Test
- 0547 – NIT Technical Decision for Clarification on developer disclosure of AVA_VAN
- 0546 – NIT Technical Decision for DTLS - clarification of Application Note 63
  - N/A to the TOE; DTLS functionality is not claimed
- 0538 – NIT Technical Decision for Outdated link to allowed-with list
  - N/A to the TOE; DTLS functionality is not claimed
- 0537 – NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3
- 0536 – NIT Technical Decision for Update Verification Inconsistency
- 0528 – NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4
  - N/A to the TOE; FCS_NTP_EXT.1 is not claimed
- 0527 – Updates to Certificate Revocation Testing (FIA_X509_EXT.1)

- 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

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash").
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). Note that 'cases' that are not applicable in a given SFR have simply been removed without any explicit identification.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.3.1 Terminology

This section identifies TOE-specific terminology.

CAVP	Cryptographic Algorithm Validation Program
GbE	Gigabit Ethernet
ICA	Independent Computing Architecture
IPMI	Intelligent Platform Management Interface
LAN	Local Area Network
RDP	Remote Desktop Protocol
SCP	Secure Copy
SFTP	SSH File Transfer Protocol
SPP	Safeguard for Privileged Passwords
USB	Universal Serial Bus
VGA	Video Graphics Array

1.3.2 Abbreviations

This section identifies abbreviations and acronyms used in this ST.

AAA	Authentication, Authorization, and Accounting
AD	Active Directory
AES	Advanced Encryption Standard
CRL	Certificate Revocation List
CSP	Critical Security Parameter
DH	Diffie-Hellman
FIPS	Federal Information Processing Standard
GPG	GnuPrivacy Guard
GUI	Graphical User Interface
HMAC	Hashed Message Authentication Code
HTTP(S)	Hypertext Transfer Protocol (Secure)
LDAP	Lightweight Directory Access Protocol
OE	Operational Environment
OS	Operating System
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SPS	Safeguard for Privileged Sessions (the TOE)
SSH	Secure Shell
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functions
VNC	Virtual networking connections

2. Product and TOE Description

Section 2.1 describes the SPS solution as a whole and the remaining subsections describe the evaluated TOE and TOE functionality included in the scope of evaluation.

2.1 Product Overview

This sub-section provides an overview of the capabilities of the One Identity SPS solution.

Safeguard for Privileged Sessions is a network appliance that is able to enforce access control, authorization, and accounting methods on application-layer protocols that are commonly associated with management activities, such as RDP, SSH and Telnet. It is configured to intercept communications between clients and servers and enforce policies based on the actions being performed. For example, a user may attempt to access an administrator account on a server over RDP (Remote Desktop Protocol). SPS can be configured to intercept this attempt and serve an additional login prompt to require the user to authenticate themselves against AD in addition to entering the administrator password. Once the user has authenticated to the target system, SPS can intercept their activities and evaluate them against policies, which determine whether the user's actions are allowed. For example, SPS could permit Remote Desktop management of the target system but prohibit any attempted use of the clipboard on that system. The product can also generate audit logs and real-time video of the actions being performed against target systems.

SPS also works with encrypted communications by supporting break-and-inspect capability. It does this by establishing itself as a proxy server and positioning itself as the server for any SSH/TLS connections that match the target policies. When it intercepts an encrypted connection, it will decrypt it, perform whatever inspection operations are needed, re-encrypt it, and send the traffic to its original destination. The product can be configured in both transparent and non-transparent modes. In transparent mode, the user attempts to connect directly to the target system and SPS is intercepting it. In non-transparent mode, the user connects directly to SPS to choose a target system. Note however that while the product has the ability to decrypt and re-encrypt TLS sessions, it is not intended for deployment as an "SSL/TLS inspection proxy" product because the purpose of such a product is to monitor end user activity in a way that maximizes compatibility with third-party websites or other TLS servers. Safeguard for Privileged Sessions is intended to be used as an application-layer access control mechanism for a known set of organizational assets; as such, it deliberately does not support the full range of TLS cipher suites expected of an SSL/TLS inspection proxy because this range includes cipher suites that are typically understood to be insecure and would not be enabled on the TLS servers that the product facilitates connectivity with.

SPS examines network traffic at the application level, i.e., Layer 7 of the OSI model. All communication must conform to the standards of the respective protocol. SPS examines Secure Shell (SSH), Secure Copy (SCP), SSH File Transfer Protocol (SFTP), Remote Desktop (RDP), HTTP, Independent Computing Architecture (Citrix ICA), Telnet, and VNC connections, ignoring and simply forwarding all other types of traffic. SPS uses man-in-the-middle techniques to decrypt and terminate (when necessary) the inspected connections. It separates the connections into two parts (client — SPS, SPS — server) and inspects all traffic so that no data can be directly transferred between the server and the client.

2.2 TOE Overview

Safeguard for Privileged Sessions is a network appliance that is able to enforce access control, authorization, and accounting methods on application-layer protocols that are commonly associated with management activities. In the evaluated configuration, the TOE is responsible for secure proxying of SSH connections that carry application-layer protocols; the access control functionality for application-layer protocols is out of scope. Specifically, the TOE is responsible for ensuring the security of its own use and for the proper implementation of the secure communications protocols used for communication to, from, and through it.

The focus of this evaluation is on the TOE functionality supporting the claims in the collaborative Protection Profile for Network Devices ([NDcPP] – see Section 1.2 for specific version information). The security functionality specified in the NDcPP includes protection of communications between the TOE and external IT entities, identification and authentication of administrators, auditing of security-relevant events, ability to verify the source and integrity of updates to the TOE, and use of NIST-validated cryptographic mechanisms.

2.3 TOE Architecture

The TOE is a standalone network device consisting of a hardware appliance (3000/3500) installed with the SPS v6.9 software/firmware. The SPS software/firmware is based on Ubuntu Linux LTS 18.04, hardened according to the official Ubuntu guides. The appliance is installed in-line between clients and servers to facilitate communications and enforce access control, authorization, and accounting methods on application-layer protocols. The TOE supports local and remote administration.

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

Note that the original tested version of the TOE was version 6.9.3; the final tested version of the TOE was 6.9.4 to include minor product enhancements needed to conform to the requirements claimed in this ST. All security-relevant administrative guidance for version 6.9.3 applies to version 6.9.4 as-is.

2.4 Physical Boundaries

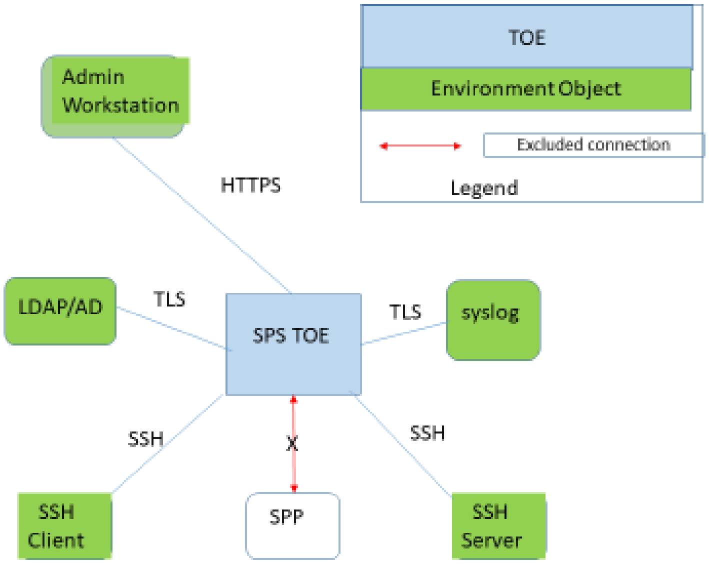
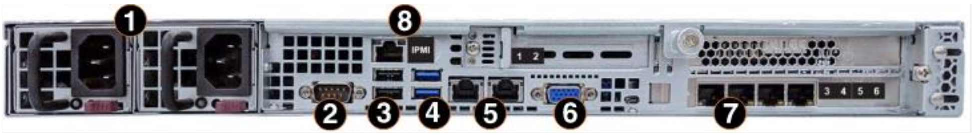


Figure 1: TOE in Operational Environment

2.4.1 TOE Components

The TOE is provided as a hardware network appliance pre-installed with software/firmware as identified in Section 1.1.

Model 3000:

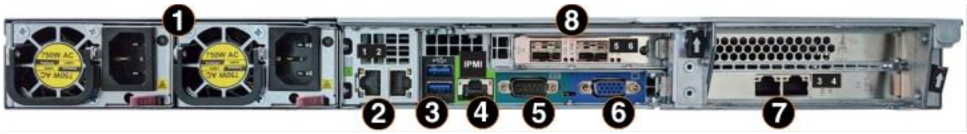


1. Redundant power supplies
2. Serial port

3. 2x USB ports
4. 2x USB ports
5. 2x RJ45 GbE Ethernet ports
6. VGA port
7. 4 x 1GBase-T Ethernet ports
8. Dedicated IPMI LAN port

The Model 3000 has an Intel Xeon E3-1275 v6 3.80 Ghz 4 Core CPU with 2x 16GB memory and 6TB usable storage.

Model 3500



1. Redundant power supplies
2. 2x RJ45 10GbE Ethernet ports
3. 2x 3.0 USB ports
4. Dedicated IPMI LAN port
5. Serial port
6. VGA port
7. 2 x 1GBase-T Ethernet ports
8. 2x SFP+ 10GbE ports

The Model 3500 has 2 x Intel Xeon Silver 4110 2.1 Ghz 8 Core CPU with 8X8 GB memory and 12TB usable storage.

The appliances differ in their performance capability but offer equivalent security functionality.

2.4.1.1 Operational Environment Components

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

- A TLS-protected syslog server that receives audit events from the TOE,
- An external LDAP/AD Server for authentication,
- At least one SSH client and at least one SSH server for session proxying using SSH, and
- A client workstation for administrator access to the web UI with:
  - 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. The OS must have the ability to run fairly recent versions of the admin clients (e.g. SSH) and a recent browser.
  - A supported browser: current version of Mozilla Firefox, current version of Google Chrome, Microsoft Edge, and Microsoft Internet Explorer 11 or newer. The browser must support TLS-encrypted HTTPS connections, and JavaScript/cookies must be enabled.

2.4.2 Excluded Components/Functionality

The list below identifies features or protocols that are not evaluated or must be disabled, and the rationale why. Note that this does not mean the features cannot be used in the evaluated configuration (unless explicitly stated so). It means that the features were not evaluated and/or validated by an independent third party and the functional correctness of the implementation is vendor assertion. Evaluated functionality is scoped exclusively to the security functional requirements specified in Security Target. In particular, only the following protocols implemented by the TOE have been tested, and only to the extent specified by the security functional requirements: TLS, HTTPS, and SSH. The features below are out of scope.

Feature	Description
Cluster and HA Deployments	Cluster and HA deployments were not evaluated.
Desktop Player Application	Excluded from the evaluation boundary since its functionality does not relate to the NDcPP requirements.
Privileged Analytics	Privileged Analytics is enabled for the evaluated configuration of the product but is outside the evaluation boundary since its functionality does not relate to the NDcPP requirements. Enabling this feature shows that it is “non-interfering” with respect to the product’s ability to meet the claimed security requirements
RADIUS	RADIUS must be disabled in the evaluated configuration because it does not use a cryptographic channel (e.g. RadSec).
Safeguard for Privileged Passwords (SPP)	Use of SPP for any purpose (i.e. user-initiated and SPS-initiated) is excluded. The channel must not be configured/used.
TLS break-and-inspect functionality (TLS/HTTPS proxy)	Excluded from the evaluated configuration since its functionality is not related to the NDcPP requirements.
SSH administrative access	Administrative access using SSH is disabled in the evaluated configuration.
Any features not associated with SFRs in [NDcPP]	NDcPP forbids adding additional requirements to the Security Target (ST). If additional functionalities or products are mentioned in the ST, it is for completeness only.

Table 1 Excluded Components/Features

2.4.3 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

2.4.3.1 Security Audit

SPS 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.

2.4.3.2 Cryptographic Support

The uses OpenSSL with NIST-validated algorithm implementations in support of its cryptographic functions. The TOE uses these algorithms to implement TLS, HTTPS, and SSH in accordance with defined standards.

2.4.3.3 Identification and Authentication

The TOE provides identification and authentication and password management functions for its administrative interface. It also supports X.509 certificate services in support of authentication for cryptographic channels.

2.4.3.4 Security Management

The TOE provides security management functions and defines roles that can be associated with users in order to manage the TOE locally or remotely. The management functions are provided through a Web UI, REST API, and local Console.

2.4.3.5 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 its own timing mechanism 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, maintenance state. It also includes a mechanism to verify TOE updates to prevent malicious or other unexpected changes in the TOE.

2.4.3.6 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.

2.4.3.7 Trusted Path/Channels

The TOE protects interactive communication with remote administrators using HTTPS (TLSv1.1 and TLSv1.2). TLS ensures both integrity and disclosure protection when administrators access the web UI and REST APIs remotely.

The TOE implements both SSH and TLS in support of communications between itself and various external entities in its operational environment.

2.5 TOE Documentation

The following product documentation applies to the TOE:

- One Identity Safeguard for Privileged Sessions 6.9 Common Criteria Evaluated Configuration Guide (CCECG), Version 1.1, 1/26/22
- One Identity Safeguard for Privileged Sessions 6.9.3 Installation Guide, April 30, 2021
- One Identity Safeguard for Privileged Sessions 6.9.3 Packaging Checklist, April 30, 2021
- One Identity Safeguard for Privileged Sessions 6.9.3 Upgrade Guide, April 30, 2021
- One Identity Safeguard for Privileged Sessions 6.9.3 Administration Guide, April 30, 2021
- One Identity Safeguard for Privileged Sessions 6.9.3 REST API Reference Guide, April 30, 2021
- Super SC113 Chassis Series User's Manual, Version 1.0d, October 2, 2013
- Supermicro SuperServer 1029U-T Series User's Manual, Revision 1.0i, September 22, 2020
- One Identity Safeguard for Privileged Sessions 6.9.4 Release Notes, January 27, 2022
- Supermicro BMC IPMI User's Guide, Revision 1.1b, August 26, 2020
- One Identity Safeguard for Privileged Sessions 6.9.4 Release Notes, January 27, 2022 (consulted to show that the changes between version 6.9.3 and 6.9.4 have no functional impact on the vendor administrative guidance)

3. Security Problem Definition

This security target includes by reference the Security Problem Definition (composed of organizational policies, threat statements, and assumptions) from the NDcPP.

In general, the NDcPP has presented a Security Problem Definition appropriate for network infrastructure devices, and as such is applicable to the One Identity SPS TOE.

Note however that the NDcPP includes a number of assumptions that only apply to products with certain characteristics. The following are not applicable to the TOE:

- A.COMPONENTS_RUNNING – NDcPP states that this only applies to distributed TOEs and the TOE is standalone.
- A.VS_TRUSTED_ADMINISTRATOR – NDcPP states that this only applies to virtual network devices and the TOE is physical.
- A.VS_REGULAR_UPDATES – NDcPP states that this only applies to virtual network devices and the TOE is physical.
- A.VS_ISOLATION– NDcPP states that this only applies to virtual network devices and the TOE is physical.
- A.VS_CORRECT_CONFIGURATION – NDcPP states that this only applies to virtual network devices and the TOE is physical.

4. Security Objectives

Like the Security Problem Definition, this security target includes by reference the Security Objectives from the NDcPP. The NDcPP security objectives for the operational environment are reproduced below, since these objectives characterize technical and procedural measures each consumer must implement in their operational environment.

In general, the NDcPP has presented a Security Objectives statement appropriate for network infrastructure devices, and as such is applicable to the One Identity SPS TOE.

Note however that the NDcPP includes a number of objectives that only apply to products with certain characteristics. The following are not applicable to the TOE:

- OE.COMPONENTS_RUNNING – NDcPP states that this only applies to distributed TOEs and the TOE is standalone.
- OE.VM_CONFIGURATION – NDcPP states that this only applies to virtual network devices and the TOE is physical.

4.1 Security Objectives for the Operational Environment

OE.PHYSICAL

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

OE.NO_GENERAL_PURPOSE

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

OE.NO_THRU_TRAFFIC_PROTECTION

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

OE.TRUSTED_ADMIN

Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality.

For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE’s trust store in case such certificate can no longer be trusted.

OE.UPDATES

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

OE.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. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.

5. 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 Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the Protection Profile (PP): collaborative Protection Profile for Network Devices, Version 2.2e, 23-March-2020, NDcPP and including the following optional and selection-based SFRs: FCS_HTTPS_EXT.1, FCS_SSHC_EXT.1, FCS_SSHS_EXT.1, 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, and FMT_MTD.1/Cryptokeys.

As a result, refinements and operations already performed in that PP are not identified (e.g., highlighted) here, rather the requirements have been copied from that PP and any residual operations have been completed herein. Of particular note, the NDcPP made a number of refinements and completed some of the SFR operations defined in the CC and that PP should be consulted to identify those changes if necessary. Text deleted from SFRs by a refinement in NDcPP is not reproduced in ST.

The SARs are the set of SARs specified in NDcPP.

5.1 Extended Requirements

All extended requirements in this ST have been drawn from the NDcPP. The NDcPP defines the following extended SFRs and since they are not redefined in this ST, the NDcPP should be consulted for more information regarding those CC extensions.

- FAU_STG_EXT.1: External Audit Event Storage
- FCS_HTTPS_EXT.1: HTTPS Protocol
- FCS_RBG_EXT.1: Random Bit Generation
- FCS_SSHC_EXT.1: SSH Client Protocol
- FCS_SSHS_EXT.1: SSH Server Protocol
- FCS_TLSC_EXT.1: TLS Client Protocol
- FCS_TLSC_EXT.2: TLS Client Protocol with mutual authentication
- FCS_TLSS_EXT.1: TLS Server Protocol
- FIA_PMG_EXT.1: Password Management
- FIA_UIA_EXT.1: User Identification and Authentication
- FIA_UAU_EXT.2: Password-based Authentication Mechanism
- FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- FIA_X509_EXT.2: X509 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 (for Reading of all Pre-shared, Symmetric and Private Keys)
- FPT_STM_EXT.1: Reliable Time Stamps
- FPT_TST_EXT.1: TSF Testing
- FPT_TUD_EXT.1: Extended: Trusted Update
- FTA_SSL_EXT.1: TSF-initiated Session Locking

5.2 TOE Security Functional Requirements

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

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
FCS: Cryptographic Support	FCS_CKM.1: Cryptographic Key Generation
	FCS_CKM.2: Cryptographic Key Establishment
	FCS_CKM.4: Cryptographic Key Destruction
	FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/ Decryption)
	FCS_COP.1/SigGen: Cryptographic Operation (Signature Generation and Verification)
	FCS_COP.1/Hash: Cryptographic Operation (Hash Algorithm)
	FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm)
	FCS_HTTPS_EXT.1: HTTPS Protocol
	FCS_RBG_EXT.1: Random Bit Generation
	FCS_SSHC_EXT.1: SSH Client Protocol
	FCS_SSHS_EXT.1: SSH Server Protocol
	FCS_TLSC_EXT.1: TLS Client Protocol
	FCS_TLSC_EXT.2: TLS Client Protocol with Mutual Authentication
	FCS_TLSS_EXT.1: TLS Server Protocol
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/ManualUpdate: Management of Security Functions Behaviour
	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_TST_EXT.1: TSF Testing
	FPT_TUD_EXT.1: Trusted Update

Requirement Class	Requirement Component
	FPT_STM_EXT.1: Reliable Time Stamps
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

Table 2 TOE Security Functional Components

5.2.1 Security Audit (FAU)

5.2.1.1 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 **3**.

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, information specified in column three of Table **3**.

Requirement	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	Failure to establish a HTTPS Session	Reason for failure
FCS_RBG_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_SSHC_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
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.
FIA_X509_EXT.3	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	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.

Requirement	Auditable Events	Additional Audit Record Contents
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions	None.

Table 3 Auditable Events

5.2.1.2 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.

5.2.1.3 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 [***[prevent the execution of TSF functions]***] when the local storage space for audit data is full.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 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: **[**

- ***ECC schemes using "NIST curves" [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;***
- ***FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1***
- ***FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526].***

].

5.2.2.2 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 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";***
- ***Finite field-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";***

- [1]***FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [groups listed in RFC 3526,]***

].

5.2.2.3 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 [***instructs a part of the TSF to destroy the abstraction that represents the key***]

that meets the following: No Standard.

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

FCS_COP.1.1/DataEncryption The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC, CTR, GCM***] mode and cryptographic key sizes [***128 bits, 256 bits***] that meet the following: AES as specified in ISO 18033-3, [***CBC as specified in ISO 10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772***].

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

FCS_COP.1.1/SigGen The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [

- ***RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits],***
- ***Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits, 384 bits, 521 bits]***

] that meet the following: [

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

].

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

FCS_COP.1.1/Hash The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm **[*SHA-1, SHA-256, SHA-384, SHA-512*]** and message digest sizes [***160, 256, 384, 512***] bits that meet the following: ISO/IEC 10118-3:2004.

[1] Modified by TD0580

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

FCS_COP.1.1/KeyedHash The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512***] 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”.

5.2.2.8 HTTPS Protocol (FCS_HTTPS_EXT.1)

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

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

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

5.2.2.9 Random Bit Generation (FCS_RBG_EXT.1)

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

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

5.2.2.10 SSH Client Protocol (FCS_SSHC_EXT.1)

FCS_SSHC_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, [***4344, 5656, 6668, 8268***].

FCS_SSHC_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [***password-based***].

FCS_SSHC_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [**131072**] bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [***aes128-ctr, aes256-ctr***].

FCS_SSHC_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [***ssh-rsa, ecdsa-sha2-nistp256***] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHC_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [***hmac-sha2-256, hmac-sha2-512***] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.7 The TSF shall ensure that [***diffie-hellman-group14-sha1, diffie-hellman-group15-sha512, diffie-hellman-group16-sha512***] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.8 The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

FCS_SSHC_EXT.1.9 The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key and [***no other methods***] as described in RFC 4251 section 4.1.

5.2.2.11 SSH Server Protocol (FCS_SSHS_EXT.1)

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [***4344, 5656, 6668, 8268***].

FCS_SSHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [***password-based***].

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [***131072***] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [***aes128-ctr, aes256-ctr***].

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [***ssh-rsa, ecdsa-sha2-nistp256***] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [***hmac-sha2-256, hmac-sha2-512***] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [***diffie-hellman-group14-sha1, diffie-hellman-group15-sha512, diffie-hellman-group16-sha512***] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

5.2.2.12 TLS Client Protocol (FCS_TLSC_EXT.1)

FCS_TLSC_EXT.1.1 The TSF shall implement [***TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246***
- ***TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288***
- ***TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289***

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

and no other ciphersuites.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches [***the reference identifier per RFC 6125 section 6***].

FCS_TLSC_EXT.1.3 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [

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

].

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

5.2.2.13 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.

5.2.2.14 TLS Server Protocol (FCS_TLSS_EXT.1)

FCS_TLSS_EXT.1.1 The TSF shall implement [***TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- ***TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492***
- ***TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246***
- ***TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246***
- ***TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288***
- ***TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289***]

and no other ciphersuites**.**

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [***none***].

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

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

5.2.3 Identification and Authentication (FIA)

5.2.3.1 Authentication Failure Management (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [**1-50**] 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 [unlock action] is taken by an Administrator; prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until an Administrator defined time period has elapsed***].

5.2.3.2 Password Management (FIA_PMG_EXT.1)

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [***“!”, “"”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [“#”, ”+”, “;”, “<”, “=”, “>”, “[“, “\”, “]”, “^”, “`”, “{“, “|”, “}”, “_”, “.”, “/”, “:”, “?”, “-“, “ ”]***];
b) Minimum password length shall be configurable to between [**1**] and [**99**] characters**.**

5.2.3.3 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.

5.2.3.4 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.

5.2.3.5 User Identification and Authentication (FIA_UIA_EXT.1)

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

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

FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative/ user.

5.2.3.6 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.

5.2.3.7 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***].

5.2.3.8 X.509 Certificate Requests (FIA_X509_EXT.3) (Selection)

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.

5.2.4 Security Management (FMT)

5.2.4.1 Management of Functions in TSF (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.

5.2.4.2 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.

5.2.4.3 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.

5.2.4.4 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 [***hash comparison***] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;

  [
  - ***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 manage the cryptographic keys;***
  - ***Ability to configure the cryptographic functionality;***
  - ***Ability to set the time which is used for time-stamps;***
  - ***Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchors;***
  - ***Ability to import X.509v3 certificates to the TOE's trust store***

].

5.2.4.5 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.

5.2.5 Protection of the TSF (FPT)

5.2.5.1 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.

5.2.5.2 Protection of TSF Data (for Reading of all Pre-shared, 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.

5.2.5.3 Reliable Time Stamps (FPT_STM_EXT.1)

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

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

5.2.5.4 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), periodically during normal operation***] to demonstrate the correct operation of the TSF: [**software/firmware integrity, prng test**].

5.2.5.5 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 [***the most recently installed version of the TOE firmware/software***].

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

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

5.2.6 TOE Access (FTA)

5.2.6.1 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.

5.2.6.2 User-initiated Termination (FTA_SSL.4)

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

5.2.6.3 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.

5.2.6.4 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.

5.2.7 Trusted Path/Channels (FTP)

5.2.7.1 FTP_ITC.1 Inter-TSF Trusted Channel (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall be capable of using [***TLS, SSH***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***authentication server, [SSH proxy]***] 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 [**remote audit storage, administrator authentication, session proxying**].

5.2.7.2 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.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are included by reference from the NDcPP.

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

Table 4 Security Assurance Components

Consequently, the evaluation activities specified in NDcPP for the SFRs that the TOE claim all apply to the TOE evaluation.

6. TOE Summary Specification

This chapter describes the security functions:

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

6.1 Security Audit

SPS 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.

6.1.1 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), administrator login/logout, any use of an administrator command via the Web Interface, or REST API, changes to key data, and password changes as well as all of the events identified in Table 3.

All log records include the following contents: date/time, event type, user ID (i.e., username, IP address) or component (i.e., ssh, 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. Additionally and based on the event, the description of the event will include additional information as required in Table 3.

6.1.2 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.

6.1.3 FAU_STG_EXT.1: Protected Audit Event Storage

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.

Configuration changes are stored indefinitely on the file system. All other audit data is stored in a local syslog and retained for seven days. Storage space is not specifically allocated to logs, logs are written onto the main partition mounted under /mnt/firmware. The minimum size of the partition is 10TB, which varies by platform. Each day has its own associated log file so when a new day has rolled over, the oldest file is deleted and a new one is created for the new day (24 hour rotation period). If there is insufficient local storage space for audit data, the TOE prevents the execution of TSF functions and audit logs are not generated.

The TOE does not offer interfaces to modify or delete audit records. Security Administrators and users with the usergroup **basic-view** can view the local audit records after successfully authenticating.

6.2 Cryptographic Support

The TOE has obtained NIST CAVP validation for all algorithm implementations in its embedded OpenSSL 1.1.1.

Functions	Standards	Certificates
Asymmetric key generation (FCS_CKM.1)		
ECC Schemes (ECDSA P-256, P-384, P-521 curves)	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4	ECDSA #C2178
FFC Schemes (DSA 2048 bits)	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.1	DSA #C2178
FFC Schemes using ‘safe-prime’ groups and RFC 3526	NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography	N/A
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 #C2178
Finite field-based scheme	NIST Special Publication 800-56A Revision 2, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography	KAS FFC #C2178

Functions	Standards	Certificates
FFC Schemes using “safe-prime” groups	NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and groups listed in RFC 3526	N/A
Encryption/Decryption (FCS_COP.1/Data Encryption)		
AES in CBC mode (128, 256 bits)	CBC as specified in ISO 10116	AES #C2178
AES in CTR mode (128, 256 bits)	CTR as specified in ISO 10116	AES #C2178
AES in GCM mode (128, 256 bits)	GCM as specified in ISO 19772	AES #C2178
Cryptographic signature services (Signature Generation and Verification) (FCS_COP.1/SigGen)		
RSA Digital Signature Algorithm (rDSA) (2048 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 #C2178
ECDSA Elliptic Curve Digital Signature Algorithm (P-256, P-384, P-521 curves)	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” P-256, P-384, P-521; ISO/IEC 14888-3, Section 6.4	ECDSA #C2178
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 #C2178
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 #C2178
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 #C2178

Table 5 Cryptographic Functions

6.2.1 FCS_CKM.1: Cryptographic Key Generation

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

ECC schemes using P-256, P-384, P-521 NIST curves that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4 is for use with TLS and X.509 certificates.

FFC schemes using cryptographic key sizes of 2048-bit that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.1 is for use with TLS and X.509 certificates.

FFC Schemes using ‘safe-prime’ groups that meet the following: “NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and RFC 3526 is for used with SSH.

6.2.2 FCS_CKM.2: Cryptographic Key Establishment

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

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”.

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

FFC Schemes using “safe-prime” groups that meet the following: ‘NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and groups listed in RFC 3526.

The TOE uses the key establishment schemes as identified in Table 6.

Scheme	SFR	Service
Elliptic curve-based	FCS_TLSC_EXT.1 FCS_TLSC_EXT.2	Audit Server, Authentication Server
	FCS_TLSS_EXT.1	Administration
	FCS_SSHC_EXT.1 FCS_SSHS_EXT.1	SSH Proxy
Finite field-based	FCS_TLSC_EXT.1 FCS_TLSC_EXT.2	Audit Server, Authentication Server
	FCS_TLSS_EXT.1	Administration
FFC Schemes using “safe-prime” groups	FCS_SSHC_EXT.1 FCS_SSHS_EXT.1	SSH Proxy

Table 6 Scheme Usage

6.2.3 FCS_CKM.4: Cryptographic Key Destruction

The TOE destroys keys and critical security parameters (CSPs) when no longer required. Plaintext keys in volatile storage are destroyed by ‘explicit_bzero(3)’ call before freeing the variable (this corresponds with the “*destruction of reference to the key directly followed by a request for garbage collection*” selection in the requirement). The destruction of plaintext keys in non-volatile storage is executed by the invocation of an interface provided by operating system that instructs a part of the TSF to destroy the abstraction that represents the key. The TOE’s application programming interface provides a handle to the OS kernel which performs the zeroization. The interfaces used are identified in the table that follows (e.g., file system).

The TOE stores the Certificate files, CA-Certificate files, and Private-Key files used in secure communications in non-encrypted PEM format. The ECDSA private keys are temporarily in volatile memory during SSH/TLS handshake and are zeroized immediately following the handshake by an API call to the OS kernel. Keys and certificates that are part of the application’s configuration are stored unencrypted as part of the TOE’s XML configuration file. The files are stored on the file system and in all applicable cases the files are passed to OpenSSL via API calls that pass in the complete filename including full path. Each API call return is checked to make sure there were no errors. The cryptographic library itself does not return sensitive data values and is responsible for ensuring the memory that referenced those file contents is destroyed. User passwords for users with local authentication are stored hashed using SHA-512 and salted using 12 bytes of salt (5,000 rounds) in a database.

The TOE uses the following secret keys, private keys and CSPs.

Key/CSP	Storage Location	Zeroized upon:	Zeroized by:
ECDSA private key (SSH/TLS)	In memory (volatile RAM)	Handshake done (SSH keys used by the proxy are kept in memory)	Crypto library, EVP_PKEY_free(3)
TLS private key (for server)	On disk (non-volatile)	Command	Filesystem
TLS private key (for client)	On disk (non-volatile)	Command	Filesystem
DRBG entropy input/seed/key	On disk (non-volatile)	N/A	N/A
Administrator credentials	XML file (hashed using SHA-512 and salted using 12 bytes of salt (5,000 rounds))	Command	Filesystem
SSH server host keys (SSH proxy)	XML file (unencrypted)	Command	Filesystem
SSH client key(s)	Depending on the user configurable setting: "built-in encryption" vs "Password protected": In memory (volatile RAM) Credential Store Database (encrypted using AES-256-CBC, key is optionally encrypted by a password SHA-512 hash of password is used as key, encryption algorithm is AES-256-CBC)	Command	Database driver, filesystem
SSH KEK	There are two options for how the key encrypting key (KEK) is stored on the appliance: o built-in credential stores store the master key unencrypted as part of the configuration o password protected credential stores use password encrypted keys. In this case the master key is encrypted using one or more passwords (AES-256-CBC)	Command	Database driver, filesystem
SSH session key and session authentication key	In memory (volatile RAM)	Close of session	Proxy using explicit_bzero(3) call
TLS pre-master secret, session key, and session authentication key	In memory (volatile RAM)	Close of session	Crypto Library

Table 7 Secret keys, Private keys and CSPs

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

The TOE supports AES CBC, AES CTR, 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 CTR as specified in ISO 10116, and GCM as specified in ISO 19772.

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

The TOE supports rDSA (modulus 2048) and ECDSA with elliptical curve key sizes 256, 384 or 521 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, P-521 “NIST curves”; ISO/IEC 14888-3, Section 6.4.

6.2.6 FCS_COP.1/Hash 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-256 hash is used for software/firmware integrity self-testing.

The SHA-1/256/384 hash functions are used with the following cryptographic functions: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, and FCS_TLSS_EXT.1, and as part of HMAC and RSA digital signature creation and verification.

The SHA-1/256/512 hash functions are used with the following cryptographic functions: FCS_SSHC_EXT.1, FCS_SSHS_EXT.1, and as part of HMAC and RSA digital signature creation and verification.

6.2.7 FCS_COP.1/KeyedHash: 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.

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

Table 8 Keyed Hash Algorithm

6.2.8 FCS_HTTPS_EXT.1: HTTPS Protocol

The TOE implements HTTPS for web-based secure administrator sessions (UI and API). The HTTPS protocol complies with RFC 2818 and is implemented using TLS. Connections are not established when peer certificates presented are invalid.

6.2.9 FCS_RBG_EXT.1: Random Bit Generation

The TOE’s OpenSSL implements the ISO/IEC 18031:2011 Deterministic Random Bit Generator (DRBG) based on the AES 256 block cipher in counter mode (CTR_DRBG (AES-256)). The TOE instantiates the DRBG with maximum security strength, obtaining the 256 bits of entropy drawn from /dev/random to seed the DRBG.

/dev/random is seeded with a hardware-based noise from RDRAND as described in the proprietary Entropy Design document. The TOE uses this DRBG to generate all keys as well as to generate salts for password hashing.

6.2.10 FCS_SSHC_EXT.1: SSH Client Protocol

The TOE implements the SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, 4344, 5656, 6668, and 8268.

The TOE supports public key-based and password-based authentication methods as described in RFC 4252. In accordance with RFC 4253, packets greater than 131072 bytes in an SSH transport connection are dropped.

The TOE uses AES CTR 128/256 as the SSH transport encryption algorithms and the following SSH public-key based authentication algorithms: ssh-rsa and ecdsa-sha2-nistp256. All other algorithms are rejected and no optional characteristics are supported.

The TOE’s SSH transport implementation uses hmac-sha2-256, or hmac-sha2-512 as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s). The TOE uses only diffie-hellman-group14-sha1, diffie-hellman-group15-sha512, or diffie-hellman-group16-sha512 for its SSH key exchange methods.

The TOE ensures that within SSH connections the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. The TOE manages a tracking mechanism for each SSH session and continually checks both thresholds so that it can initiate a new key exchange (rekey) when either threshold has passed, whichever threshold occurs first. This behavior is not configurable and occurs by default.

The TOE’s SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key as described in RFC 4251 section 4.1.

SSH is used for SSH client/server proxy communications. SPS isolates the client-server connection into two separate connections, therefore the permitted algorithms can be different on the client and the server side.

6.2.11 FCS_SSHS_EXT.1: SSH Server Protocol

The TOE’s SSH Server Protocol is implemented using the same algorithms and according to the same protocols as for the SSH Client. See Section 6.2.10 above for details. The TOE’s SSH Server establishes a user identity when an SSH client presents a public key by verifying that the SSH client’s presented public key matches one that is stored within the SSH server’s authorized_keys file.

SSH is used for SSH client/server proxy communications. SPS isolates the client-server connection into two separate connections, therefore the permitted algorithms can be different on the client and the server side.

6.2.12 FCS_TLSC_EXT.1: TLS Client Protocol

The TOE implements TLS 1.2 (RFC 5246) and TLS 1.1 (RFC 4346) for communication channels between the TOE and the audit/authentication server. The TOE rejects all other TLS and SSL versions. TLS protocol versions are configurable from the **`Trust Stores – Cryptography settings`** page. TLS 1.2 is the recommended default setting, however TLS 1.1 may alternatively be selected. The TLS 1.1 setting will offer TLS version 1.1 and later versions during negotiation. The TLS 1.2 setting will only offer TLS version 1.2 during negotiation. TLS 1.0 must not be selected. Earlier versions are not supported. The TLS implementation supports the following ciphersuites:

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288

- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289.

The TOE's TLS client verifies that the presented identifier matches the reference identifier per RFC 6125 section 6. The TOE supports hostname and DN reference identifiers. IP addresses and wildcards are not supported.

When establishing a trusted channel, by default the TOE not establish a trusted channel if the server certificate is invalid and does not implement any administrator override mechanism.

By default, the TOE presents the Supported Elliptic Curves/Supported Groups extension in the Client Hello with the secp256r1 NIST curve.

6.2.13 FCS_TLSC_EXT.2: TLS Client Protocol with Mutual Authentication

The TOE supports TLS communication with mutual authentication using X.509v3 certificates for communications with the audit/authentication server. Mutual authentication for these channels is optional.

6.2.14 FCS_TLSS_EXT.1: TLS Server Protocol

The TOE implements TLS 1.2 (RFC 5246) and TLS 1.1 (RFC 4346) for communications with the web UI and rejects all other TLS and SSL versions. The TLS protocol versions are configurable from the **`TLS security settings`** page. TLS 1.2 is the recommended default setting, however TLS 1.1 may alternatively be selected. The TLS 1.1 setting will offer TLS version 1.1 and later versions during negotiation. The TLS 1.2 setting will only offer TLS version 1.2 during negotiation. TLS 1.0 must not be selected. Earlier versions are not supported. The TLS server implementation supports the same ciphersuites as those identified for the TOE's TLS Client in Section 6.2.12.

The TOE performs key establishment for TLS using 2048-bit Diffie-Hellman parameters and secp256r1 ECDHE curves. For the ECDHE and DHE cipher key exchange methods: secp256r1 and Group 14 (2048 MODP) are used respectively. The TOE's TLS server implementation does not support session tickets or session resumption.

6.3 Identification and Authentication

The TOE provides identification and authentication and password management functions.

6.3.1 FIA_AFL.1 Authentication Failure Management

The TOE provides a configurable method of authentication failure lockout for unsuccessful login attempts. The function applies to all interfaces except the local Console. After the configured number of authentication failures has been met (from 1 to 50 with default 20); the user is locked out until the configured time has elapsed. Lockout expiration time is configurable from between 1 to 720 minutes (default 10 minutes). An administrator can also unlock a user by executing the unlock command (available from the console). The function applies to both password and certificate authentication.

There can never be a case where no administrator access is available due to either permanent or temporary lockout. The local Console interface is not subject to the lockout function and is always available to the authorized administrator with root permission.

6.3.2 FIA_PMG_EXT.1: Password Management

Administrative passwords can be composed of letters (a-z, A-Z), numbers (0-9) the special characters (!"#$%&'()*+,;<=>@[\]^`{|}_./:?-) and the space character.

Minimum password length is configurable from a minimum of 1 number of characters to a maximum of 99 number of characters.

6.3.3 FIA_UIA_EXT.1: User Identification and Authentication

The TOE displays an administrator-specified warning banner and allows no other actions until the user is identified and authenticated.

The TOE supports remote administration over the network and local administration through a directly networked workstation via crossover cable, both use HTTPS. Authentication for HTTPS (web UI and REST API) is performed by username/password, either defined locally or through an LDAP/AD server. A logon is successful when the username and password provided by the user matches a defined account on the TOE or when the username in the certificate (domain name) matches the username in the local database. One authentication method is configured for all web UI/REST API users; it is not individually configurable on a per-user basis.

The TOE also supports local administration from its Console interface. Only the root user with physical access can access this interface. This interface supports local password authentication and displays the administrator-specified warning banner prior to authentication. No functions are available prior to authentication.

6.3.4 FIA_UAU_EXT.2: Password-based Authentication Mechanism

The password-based authentication method is described under FIA_UIA_EXT.1: User Identification and Authentication.

6.3.5 FIA_UAU.7: Protected Authentication Feedback

The TOE obscures feedback to the administrative user while the authentication is in progress. Password data is obfuscated as it is being entered, and authentication failure messages are generic no matter the failure (e.g. “invalid user” and “valid user but invalid password” are not two separate messages).

6.3.6 FIA_X509_EXT.1/Rev: X.509 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.
- Validate 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:
    - Certificates used for trusted updates and executable code integrity verification[2]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 [3]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.

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

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

The TOE supports certificate revocation checking using CRL. The revocation checking policy can be configured for an individual trust store by going to Settings > Trust Stores in the UI and selecting the trust store to be configured. In the evaluated configuration, the "Full" revocation check type must be selected and appropriate CRL URLs must be specified for each root and intermediate CA.

A CRL check is always performed when a certificate needs to be validated and is performed for each certificate in the path presented for the aforementioned channels. Specifically, certificate validity checking is performed when the TOE is acting as a TLS client (with AD/LDAP and syslog servers); when the TOE is acting as a TLS server and mutual authentication is configured for use; and when a new TLS client or server certificate is loaded onto the TOE.

6.3.7 FIA_X509_EXT.2: X.509 Certificate Authentication

The TOE uses X.509v3 certificates as defined by RFC 5280 to support client-side TLS mutual authentication for HTTPS and TLS connections (authentication and audit servers). The TOE will not accept the certificate when the TOE cannot establish a connection to determine the validity of a certificate. There is no administrative override, either overall or on a per-connection level.

When the TOE is validating a TLS client or server certificate, or when a certificate is loaded onto it for its own use, it uses the certificate that is presented to it for validation. When the TOE is providing its own server certificate to a TLS client, or when mutual authentication is configured and the TOE is providing its own client certificate to a TLS server, it will only have one certificate for each of those purposes.

6.3.8 FIA_X509_EXT.3: X.509 Certificate Requests

The TOE generates a Certificate Request as specified by RFC 2986 and provides the following information in the request: public key, Common Name, Organization, Organizational Unit, and Country. The Certificate Signing Request functionality is available through the API. The TOE validates the chain of certificates from the Root CA upon receiving the CA Certificate Response.

6.4 Security Management

The TOE provides security management functions and defines roles that can be associated with users in order to manage the TOE locally or remotely.

6.4.1 FMT_MOF.1/ManualUpdate: Management of Security Functions Behaviour Requests

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

6.4.2 FMT_MTD.1/CoreData: Management of TSF Data

No administrative functions can be performed at the TOE interfaces prior to initiating the identification and authentication process other than the display of the warning banner. The TOE restricts the ability to manage the TSF data to Security Administrators. Permissions for the web interface are role-based; assigned by group membership; and are either defined locally or in AD/LDAP. During initial configuration, a root user is established. This user is the sole Security Administrator with root access to the Console interface. The certificate trust store is located in access controlled storage. Certificates are protected from unauthorized access by not allowing unprivileged users access to the storage areas. A sufficiently privileged administrator can manage the certificate trust store using the web interface (UI/API).

6.4.3 FMT_MTD.1/CryptoKeys: Management of TSF Data

The TOE restrict the ability to manage the cryptographic keys to Security Administrators. In particular, the TOE provides web interfaces (UI/API) to manage the private keys for the TOE’s TLS and SSH connections.

6.4.4 FMT_SMF.1: Specification of Management Functions

The TOE is capable of performing the following management functions:

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using hash comparison capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- Ability to configure audit behaviour (i.e. changes to remote storage locations for audit);
- Ability to manage the cryptographic keys;
- Ability to configure the cryptographic functionality;
- Ability to set the time which is used for time-stamps;
- Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchors;
- Ability to import X.509v3 certificates to the TOE's trust store.

All management functions are available remotely and locally and via either the UI or the API.

The local Console menu provides access to basic configuration and management settings of SPS and is mainly used for troubleshooting purposes. The console menu provides the unlock command and the ability to change root and admin user passwords. The remaining functions are for troubleshooting purposes only.

6.4.5 FMT_SMR.2: Restrictions on Security Roles

The console interface has one default root user with full privileges over the available functions. The TOE implements role-based access control for the web interface only (not for the console). The TOE defines the following default roles/groups for the web interface:

- **basic-view**: View the settings in the **Basic Settings** menu, including the system logs of SPS. Members of this group can also execute commands on the **Troubleshooting** tab.
- **basic-write**: Edit the settings in the **Basic Settings** menu. Members of this group can manage SPS as a host.
- **auth-view**: View the names and privileges of the SPS administrators, the configured usergroups, and the authentication settings in the **AAA** menu. Members of this group can also view the history of configuration changes.
- **auth-write**: Edit authentication settings and manage users and usergroups.
- **search:** Browse and download various logs and alerts in the Search menu. The members of this group have access to the audit trail files as well. Note that to open encrypted audit trail files, the proper encryption keys are required.
- **changelog:** View the history of SPS configuration changes in the AAA > Accounting menu.
- **policies-view:** View the policies and settings in the Policies menu.
- **policies-write:** Edit the policies and settings in the Policies menu.
- **ssh-view:** View all connection and policy settings in the SSH Control menu.
- **ssh-write:** Edit all connection and policy settings in the SSH Control menu.
- **rdp-view:** View all connection and policy settings in the RDP Control menu.
- **rdp-write:** Edit all connection and policy settings in the RDP Control menu.
- **telnet-view:** View all connection and policy settings in the Telnet Control menu.
- **telnet-write:** Edit all connection and policy settings in the Telnet Control menu.
- **vnc-view:** View all connection and policy settings in the VNC Control menu.
- **vnc-write:** Edit all connection and policy settings in the VNC Control menu.
- **indexing:** Allows hosts running external indexers to access and download audit trails for automatic indexing. Note that the members of this group can access the SPS web interface as well, and download any audit trail directly.
- **ica-view:** View all connection and policy settings in the ICA Control menu.
- **ica-write:** Edit all connection and policy settings in the ICA Control menu.
- **api:** View and edit rights for the Access RPC API privilege, to access SPS.

- **http-view:** View all connection and policy settings in the HTTP Control menu.
- **http-write:** Edit all connection and policy settings in the HTTP Control menu.
- **indexer-view:** View all connection and policy settings in the Indexer menu.
- **indexer-write:** Edit all connection and policy settings in the Indexer menu.

Additional roles can be defined and there is a default admin user (configured during initial installation) that has full authority. However, there are no functions restricted to this admin and all privileges can be assigned to other users.

Members of the auth-write group, or any other group with write privileges to the AAA menu are essentially equivalent to system administrators of SPS, because they can give themselves any privilege. Users with limited rights should never have such privileges.

If a user with write privileges to the AAA menu gives himself new privileges (for example gives himself group membership to a new group), then they must log back in to the TOE web interface to activate the new privilege.

The system administrator is equivalent to the Security Administrator as defined in the PP.

6.5 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 its own timing mechanism 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, maintenance state. It also includes a mechanism to verify TOE updates to prevent malicious or other unexpected changes in the TOE.

6.5.1 FPT_APW_EXT.1: Protection of Administrator Passwords

Local administrative passwords are stored hashed using SHA-512 and salted using 12 bytes of salt (5,000 rounds). There are no administrative interfaces to view the stored password data.

6.5.2 FPT_SKP_EXT.1: Protection of TSF Data (for Reading of all Pre-shared, Symmetric and Private Keys)

The TOE does not provide interfaces that allow pre-shared, symmetric or private keys to be read. Section 6.2 describes how the pre-shared keys, symmetric keys and private keys are stored.

SSH Proxy private keys can optionally be stored encrypted in the Credential Store. Please refer to FCS_CKM.4 for the list of all key data and how they are stored.

Administrators can download the configuration bundle over the web interface which exports all configuration data, including keys. While this data may be exported in plaintext, in the evaluated configuration, this will be encrypted using GPG using AES-CBC-256 through the admin supplying a password (which is then used to derive a key). The TOE uses GPGs default Key Derivation Function (PBKDF2).

6.5.3 FPT_STM_EXT.1: Reliable Time Stamps

The TOE provides reliable time stamps for its own use and allows the Security Administrator to set the time using **Basic Settings > Date & Time**.

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 (e.g. SSH rekeying and certificate expiration check). The clock is also used for timing of administrator lockout function (FIA_AFL.1).

6.5.4 FPT_TST_EXT.1: TSF Testing

During initial start-up (on power on), the TOE runs a suite of software/firmware integrity and prng self-tests that demonstrate the correct operation of the TSF. All self-tests are performed during start-up and additionally the prng test runs every 10 seconds (periodically during normal operation). The integrity test verifies the software/firmware's sha256 checksum during initial start-up by comparing the calculated checksum hash against the expected checksum provided with the TOE files. If the integrity test fails, an error is logged and a warning message is shown to the administrator after login to the web interface. The prng test checks the random generator for repeating patterns every 10s. If repeating patterns are detected, the test fails, an error message is logged, and all "production" related services are shut down. In this manner, the self-tests are sufficient to validate the integrity of the TOE and to ensure that the TOE and its cryptographic functionality are operating correctly.

6.5.5 FPT_TUD_EXT.1: Trusted Update

The TOE provides Security Administrators with a manual trusted update mechanism used to initiate updates to TOE firmware/software. The firmware image uses a published hash (SHA-1) that is included on the download page on the One Identity Support portal and validated by the administrator prior to uploading the image.

The TOE can have up to five images loaded including the currently executing version and all versions can be queried. The current and all loaded images can be viewed from: ***Basic Settings* > *System* > *Firmwares***. From this same page, the administrator chooses from the available firmware images that have been previously loaded. However, the chosen image does not become active until the administrator reboots the TOE.

6.6 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.

6.6.1 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 720 minutes (12 hours). The default is ten minutes. One configurable setting in the web interface applies to local and remote console, web UI and the REST API.

6.6.2 FTA_SSL.4: User-initiated Termination

The TOE allows administrator-initiated termination of the administrator's own interactive session. The Web UI provides a logout option, and the REST API has the delete operation for the session.

6.6.3 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 720 minutes (12 hours). The default is ten minutes. One configurable setting in the web interface applies to local and remote console, web UI and the REST API.

6.6.4 FTA_TAB.1: Default TOE Access Banners

Before establishing an administrative user session the TOE displays an administrator-specified advisory notice and consent warning message regarding use of the TOE. The single text banner is displayed at the HTTPS and local console administrative interfaces prior to login. It does not apply to the REST API programmatic interface.

6.7 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.

6.7.1 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, administrator authentication, and session proxying using the following:

- Syslog server using TLS - the TOE acts as a client and the non-TSF endpoint is identified by X.509 certificate,
- LDAP/AD authentication servers using TLS - the TOE acts as a client and the non-TSF endpoint is identified by X.509 certificate, and
- SSH session proxying using SSH - the TOE acts as both a client and a server and the non-TSF endpoint is identified by hostname/public key (when TOE is acting as client) and by claimed subject identity (username/public key) (when TOE is acting as server).

The TOE communicates with its authorized trusted entities over TLS or SSH. All communication 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.

6.7.2 FTP_TRP.1/Admin: Trusted Path

The TOE provides HTTPS (TLSv1.1 and TLSv1.2) to support secure remote administration when administrators access the web UI and REST APIs remotely. Mutual authentication for this interface is optional. 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.

7. Protection Profile Claims

The ST conforms to the collaborative Protection Profile for Network Devices, Version 2.2e, 23-March-2020, NDcPP.

As explained in Section 3, Security Problem Definition, the Security Problem Definition of the NDcPP has been included by reference into this ST.

As explained in Section 4, Security Objectives, the Security Objectives of the NDcPP have been included by reference into this ST.

All Security Functional Requirements (SFRs) in this ST have been reproduced from the NDcPP and operations completed as appropriate.

8. Rationale

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