# Pure Storage FlashArray//CR3, //CR4, //E, //XL, //XR3, and //XR4 Appliances Running Purity 6.7 Security Target

Document Version: 1.4

**Revision History:**

| Version | Date | Changes |
|---|---|---|
| Version 1.0 | 18-Nov-2024 | Initial Draft |
| Version 1.1 | 31-Aug-2025 | Second Draft |
| Version 1.2 | 03-Oct-2025 | Third Draft |
| Version 1.3 | 30-Nov-2025 | Fourth Draft |
| Version 1.4 | 1-Dec-2025 | Fifth Draft |

# Contents

1 Introduction ..... 5
1.1 Security Target and TOE Reference ..... 5
1.2 TOE Overview ..... 5
1.3 TOE Description ..... 5
1.3.1 Physical Boundaries ..... 6
1.3.2 Security Functions Provided by the TOE ..... 8
1.3.3 TOE Documentation ..... 11
1.3.4 References ..... 11
1.4 TOE Environment ..... 11
1.5 Product Functionality not Included in the Scope of the Evaluation ..... 12
2 Conformance Claims ..... 14
2.1 CC Conformance Claims ..... 14
2.2 Protection Profile Conformance ..... 14
2.3 Conformance Rationale ..... 14
2.3.1 Technical Decisions ..... 14
3 Security Problem Definition ..... 16
3.1 Threats ..... 16
3.2 Assumptions ..... 18
3.3 Organizational Security Policies ..... 19
4 Security Objectives ..... 20
4.1 Security Objectives for the Operational Environment ..... 20
5 Security Requirements ..... 21
5.1 Conventions ..... 22
5.2 Security Functional Requirements ..... 22
5.2.1 Security Audit (FAU) ..... 22
5.2.2 Cryptographic Support (FCS) ..... 25
5.2.3 Identification and Authentication (FIA) ..... 30
5.2.4 Security Management (FMT) ..... 32
5.2.5 Protection of the TSF (FPT) ..... 34
5.2.6 TOE Access (FTA) ..... 35
5.2.7 Trusted Path/Channels (FTP) ..... 36
5.3 TOE SFR Dependencies Rationale for SFRs ..... 37
5.4 Security Assurance Requirements ..... 37

6 TOE Summary Specifications ..... 38
6.1 CAVP Algorithm Certificate Details ..... 55
6.2 Cryptographic Key Destruction ..... 56
7 Acronym Table ..... 57

# 1 Introduction

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document will also describe the intended operational environment of the TOE, and the functional and assurance requirements that the TOE meets.

## 1.1 Security Target and TOE Reference

This section provides the information needed to identify and control the TOE and the ST.

**Table 1 - TOE/ST Identification**

| Category | Identifier |
|---|---|
| ST Title | Pure Storage FlashArray//CR3, //CR4, //E, //XL, //XR3, and //XR4 Appliances Running Purity 6.7 Security Target |
| ST Version | 1.4 |
| ST Date | December 1, 2025 |
| ST Author | Pure Storage, Inc. |
| TOE Reference | Pure Storage FlashArray Appliances |
| TOE Hardware | see Section 1.3.1 |
| TOE Version | Purity 6.7.5.post2 |
| TOE Developer | Pure Storage, Inc |
| Key Words | Network Device, SSH, TLS |

## 1.2 TOE Overview

The TOE is the Pure Storage Inc (Pure Storage) FlashArray//CR3, //CR4, //E, //XL, //XR3, and //XR4 model families running Purity v6.7.

The usage and major security features of the TOE are described in Section 1.3 “TOE Description” below. The TOE Environment is described in Section 1.4 “TOE Environment”

The TOE is classified as a Network Device.

## 1.3 TOE Description

Pure Storage Inc's (Pure Storage) FlashArray (TOE) is an enterprise Network Attached Storage solution that includes a Linux-based operating system, SAN (Storage Area Network) protocols and interfaces (iSCSI, Fiber Channel, SAS), and custom software to provide network storage with high performance, reliability, usability, and efficiency. The TOE comes with the following unevaluated SAN features:

- 5-10x Data Reduction (Purity Reduce)
- Non-Disruptive Expansion and High Availability (Purity Assure)
- Snapshots, Backup & Disaster Recovery (Purity Protect)
- Real-world Optimized Performance (100K - 200K 32K IOPS @ <1ms average latency)
- Data at rest encryption with AES-256 (Purity Secure)

The Pure Storage FlashArray is designed to act as a data storage endpoint for a SAN (the data stored as part of SAN operations is not considered to be TSF data). The TOE supports remote administration over SSH with cryptographic encryption and authentication using FIPS 140-2 approved algorithms. The TOE also supports use of external audit servers, protected by TLS.

The TOE is not a distributed TOE.

### 1.3.1 Physical Boundaries

The physical boundaries of the TOE consist of the physical appliance including all the hardware and the software. The TOE appliance model numbers and corresponding processor are shown in the table below.

**Table 2 - TOE Hardware Models**

| Model # | Processor | | CPU Microarchitecture |
|---|---|---|---|
| Flash Array X10 R3 | Intel Xeon Silver 4208 | | Cascade Lake |
| Flash Array X20 R3 | Intel Xeon Silver 4210R | | Cascade Lake |
| Flash Array X50 R3 | Intel Xeon Silver 4214R | | Cascade Lake |
| Flash Array X70 R3 | Intel Xeon Gold 6230 | | Cascade Lake |
| Flash Array X90 R3 | Intel Xeon Silver 6252 | | Cascade Lake |
| FlashArray X20 R4 | Intel Xeon Silver 4410Y | | Sapphire Rapids |
| FlashArray X50 R4 | Intel Xeon Silver 4410Y | | Sapphire Rapids |
| FlashArray X70 R4 | Intel Xeon Gold 5416S | | Sapphire Rapids |
| FlashArray X90 R4 | Intel Xeon Gold 5418N | | Sapphire Rapids |
| FlashArray C60 R3 | Intel Xeon Gold 6230 | | Cascade Lake |
| FlashArray C40 R3 | Intel Xeon Silver 4210R | | Cascade Lake |
| FlashArray C50 R4 | Intel Xeon Silver 4410Y | | Sapphire Rapids |
| FlashArray C70 R4 | Intel Xeon Gold 5416S | | Sapphire Rapids |
| FlashArray C90 R4 | Intel Xeon Gold 5418N | | Sapphire Rapids |
| FlashArray E | Intel Xeon Silver 4410Y | | Sapphire Rapids |
| FlashArray XL130 | Intel Xeon Gold 6338 | | Ice Lake |
| FlashArray XL170 | Intel Xeon Platinum 8368 | | Ice Lake |

The physical boundaries of the TOEs are illustrated in the figure below with red dotted lines.

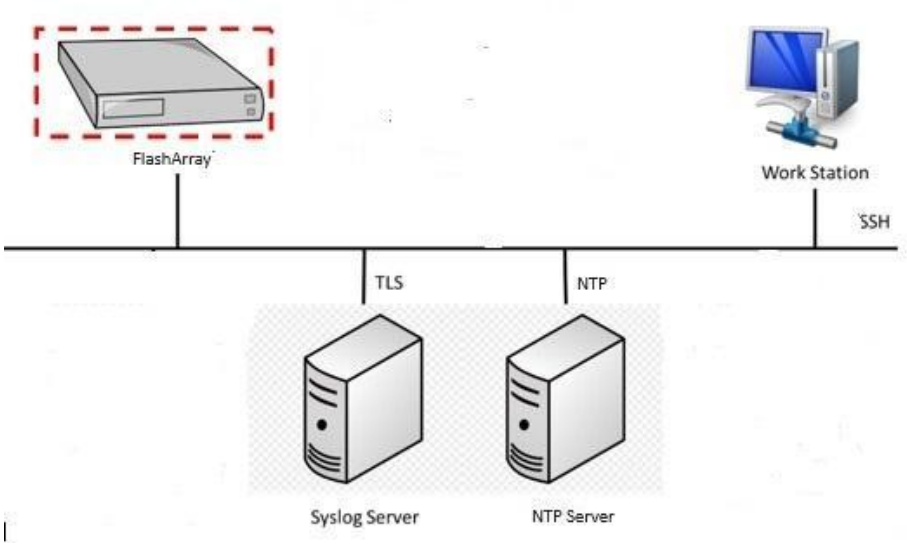


**Figure 1 – TOE Boundaries**

Summary specifications for the TOE appliance models is provided in the following tables.

**Table 3 - FlashArray//X R4 & FlashArray//C R4 Family models and specifications**

| Model | X20 R4 | X50 R4 / C50 R4 | X70 R4 / C70 R4 | X90 R4 / C90 R4 |
|---|---|---|---|---|
| **CPU** | Intel® Xeon® Silver 4410Y Processor | Intel® Xeon® Silver 4410Y Processor | Intel® Xeon® Gold 5416S Processor | Intel® Xeon® Gold 5418N Processor |
| **Total Volatile Memory** | 256 GB | 384 GB | 512 GB | 1024 GB |
| **Management Ports** | 2 x 1Gb | 2 x 1Gb | 2 x 1Gb | 2 x 1Gb |
| **Local Console Ports** | 1x | 1x | 1x | 1x |

**Table 4 - FlashArray//X R3 Family models and specifications**

| | X10 R3 | X20 R3 | X50 R3 | X70 R3 | X90 R3 |
|---|---|---|---|---|---|
| **CPU** | Intel® Xeon® Silver 4208 Processor | Intel® Xeon® Silver 4210R Processor | Intel® Xeon® Silver 4214R Processor | Intel® Xeon® Gold 6230 Processor | Intel® Xeon® Silver 6252 Processor |
| **Total Volatile Memory** | 96 GB | 192 GB | 288 GB | 384 GB | 768 GB |
| **Management Ports** | 2 x 1Gb | 2 x 1Gb | 2 x 1Gb | 2 x 1Gb | 2 x 1Gb |
| **Local Console Ports** | 1x | 1x | 1x | 1x | 1x |

**Table 5 - FlashArray//C R3, FlashArray//XL, & FlashArray//E Family models and specifications**

| | C60 R3 | C40 R3 | XL130 | XL170 | E |
|---|---|---|---|---|---|
| **CPU** | Intel Xeon Gold 6230 Processor | Intel Xeon Silver 4210R Processor | Intel® Xeon® Gold 6338 Processor | Intel® Xeon® Platinum 8368 Processor | Intel® Xeon® Silver 4410Y Processor |
| **Total Volatile Memory** | 768 TB | 384 TB | 768 GB | 1 TB | 1 TB |
| **Management Ports** | 2 x 1Gb | 2 x 1Gb | 2 x 1Gb | 2 x 1Gb | 2 x 1Gb |
| **Local Console Ports** | 1x | 1x | 1x | 1x | 1x |

### 1.3.2 Security Functions Provided by the TOE

The TOE provides the security functions required by NDcPP v3.0e [PP-ND] and PKG_SSH v1.0 [PKG-SSH].

#### 1.3.2.1 Security Audit

- The TOE generates and stores audit events locally and forwards the logs remotely to syslog server securely via TLS.
- The TOE will audit all events and information defined in Table 12: Auditable Events.
- The TOE will also include the identity of the user that caused the event (if applicable), date and time of the event, type of event, and the outcome of the event.
- The TOE protects storage of audit information from unauthorized deletion.
- The TOE prevents unauthorized modifications to the stored audit records.

#### 1.3.2.2 Cryptographic Support

The TSF performs the following cryptographic operations:

- SSH for remote CLI administrative management of the TOE:
  - Protocol versions:
    - SSHv2 (Conforming to RFCs 4251 - 4254, 4344, 5656, 6668, 8308, 8332)
  - Public-Key Algorithms:
    - rsa-sha2-256 - 2048 bit key
    - rsa-sha2-512 - 2048 bit key
  - Data Encryption - Algorithms:
    - aes128-cbc - 128-bit AES symmetric key
    - aes256-cbc - 256-bit AES symmetric key
    - aes128-ctr - 128-bit AES symmetric key
    - aes256-ctr - 256-bit AES symmetric key
    - aes128-gcm@openssh.com - 128-bit AES symmetric key
    - aes256-gcm@openssh.com - 256-bit AES symmetric key
  - Data Encryption - MACs:
    - hmac-sha2-256, hmac-sha2-512
  - Key Exchange:
    - ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521
- Syslog Server:
  - Conforming to the following RFCs:
    - RFC 3164 - The BSD syslog Protocol, RFC 5425 - TLS Transport Mapping
  - Supporting the following for TLS:
    - TLSv1.2 (Conforming to RFC 5246)
  - Supporting at least one of the following TLS Ciphersuites:
    - TLS_RSA_WITH_AES_128_CBC_SHA256
    - TLS_RSA_WITH_AES_256_CBC_SHA256

        - TLS_RSA_WITH_AES_128_GCM_SHA256
        - TLS_RSA_WITH_AES_256_GCM_SHA384
        - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
        - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
        - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
        - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
        - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
        - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
        - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
        - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

- NTP Server:
    - Conforming to one of the following RFCs:
        - RFC 5905 - Network Time Protocol Version 4
    - Supporting one of the following NTP versions:
        - NTP v4
    - Supporting authentication using messages digests with the SHA-256 algorithm

The TSF zeroizes all plaintext secret and private cryptographic keys and CSPs once they are no longer required.

#### 1.3.2.3 Identification and Authentication

The TSF supports passwords consisting of alphanumeric and special characters.

- The TSF allows the security administrator to configure the minimum password length from 1 character to 100 characters.
- The TSF prevents offending Administrator accounts (FIA_AFL.1.1) from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed
- The TSF allows local administrators to re-enable user accounts locked by the FIA_AFL.1 functionality
- The TSF requires all administrative-users to authenticate before allowing the user to perform any actions other than:
    - Display the warning banner in accordance with FTA_TAB.1;
    - Respond to ICMP Echo Request
    - Respond to ARP requests with ARP replies
    - Make DNS Requests

#### 1.3.2.4 Security Management

- TSF data includes the following:
    - All audit records generated to meet the auditing requirements of the PPs
    - All user credentials (symmetric keys, private keys, keying material, username/password)
    - TSF Configuration data
- The TSF includes four administrative roles within the Authorized Administrator role:
    - Internal Administrator

  - Array Administrator
  - Storage Administrator
  - Read-Only Administrator
- All roles are considered authorized administrators for the remainder of this document.
- The device ships with three hard-coded users but allows for additional users to be created.
- The TOE provides management over SSH (remote) and a local console.
- The TOE authenticates administrative users using a username/password combination or a username/SSH_RSA or username/SSH_ECDSA key combination.
- The TSF does not allow access to any administrative functions prior to successful authentication.
- The TOE also has the capability of being updated and verifying updates via digital signature.

#### 1.3.2.5 Protection of the TSF

- The TSF protects TSF data from disclosure when the data is transmitted between administrators and the TOE, and between the TOE and trusted IT entities.
- The TSF prevents the reading of secret and private keys.
- The TOE provides reliable time stamps for itself.
- The TOE runs a suite of self-tests during the initial start-up (upon power on) to demonstrate the correction operation of the TSF.

#### 1.3.2.6 TOE Access

- The TOE, for local interactive sessions, terminates the user's session after an Authorized Administrator-specified period of session inactivity (applies to the local console).
- The TOE terminates a remote interactive session after an Authorized Administrator-configurable period of session inactivity (applies to SSH remote console)
- The TOE allows Administrator-initiated termination of the Administrator's own interactive session.
- Before establishing an administrative user session, the TOE is capable of displaying an Authorized Administrator-specified advisory notice and consent warning message regarding unauthorized use of the TOE.

#### 1.3.2.7 Trusted Path/Channels

The TOE supports TLSv1.2 to secure remote communications. TLS protocol may be used for communications with remote IT entities. Remote administration is only supported using SSH.

- The TOE uses TLS to provide a trusted communication channel between itself and all authorized IT entities that are 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.
- The TOE permits the TSF, or the authorized IT entities, to initiate communication via the trusted channel.
- The TOE permits remote administrators to initiate communication via the trusted path. The TOE provides an SSH protected trusted path to administer the TOE.
- The TOE requires the use of the trusted path for initial administrator authentication and all remote administration actions.

### 1.3.3 TOE Documentation

The following documents are essential to understanding and controlling the TOE in the evaluated configuration:

- Pure Storage FlashArray//CR3, //CR4, //E, //XL, //XR3, and //XR4 Appliances Running Purity 6.7 Security Target [ST]
- Pure Storage FlashArray//CR3, //CR4, //E, //XL, //XR3, and //XR4 Appliances Running Purity 6.7 Guidance Document [ADG]

### 1.3.4 References

In addition to TOE documentation, the following references may also be valuable when understanding and controlling the TOE:

- collaborative Protection Profile for Network Devices, Version 3.0e, 6 December 2024 [PP-ND]
- Functional Package for Secure Shell (SSH) Version 1.0, 13 May2021 [PKG-SSH].

## 1.4 TOE Environment

The following environmental components are required to operate the TOE in the evaluated configuration:

Syslog Server:

- Conforming to the following RFCs:
  - RFC 3164 - The BSD syslog Protocol
  - RFC 5425 - TLS Transport Mapping

- Supporting the following for TLS:
  - TLSv1.2 (Conforming to RFC 5246)

- Supporting at least one of the following TLS Ciphersuites:
  - TLS_RSA_WITH_AES_128_CBC_SHA256
  - TLS_RSA_WITH_AES_256_CBC_SHA256
  - TLS_RSA_WITH_AES_128_GCM_SHA256
  - TLS_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
  - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
  - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
  - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

NTP Server:

- Conforming to one of the following RFCs:
  - RFC 5905 - Network Time Protocol Version 4

- Supporting one of the following NTP versions:

  - NTP v4

- Supporting authentication using messages digests with the SHA-256 algorithm

Remote Administrative User Access:

- For remote administrative user access, the TOE requires an SSH client supporting:
  - Protocol versions:
    - SSHv2 (Conforming to RFCs 4251 - 4254, 5656, and 6668, 8308, 8332)
  - Public-Key Algorithms (at least one of the following):
    - rsa-sha2-256
    - rsa-sha2-512
  - Data Encryption (at least one of the following):
    - aes128-cbc
    - aes256-cbc
    - aes128-ctr
    - aes256-ctr
    - aes128-gcm@openssh.com
    - aes256-gcm@openssh.com
  - Data Integrity (at least one of the following):
    - hmac-sha2-256,
    - hmac-sha2-512
  - Key Exchange (at least one of the following):
    - ecdh-sha2-nistp256
    - ecdh-sha2-nistp384
    - ecdh-sha2-nistp521

Hardware requirements:

- Local Console:
  - Serial I/O (SIO) Cable
  - Terminal/Monitor and Keyboard (via Terminal Emulation over SIO)

- SAS-connected SSD Storage Array from Pure Storage

## 1.5 Product Functionality not Included in the Scope of the Evaluation

The hardware and software of the TOE environment, identified above in Section 1.4, are not included in the CC evaluation scope.

Only the security functions specified in Section 5.2 (Security Functional Requirements) and Section 6 (TOE Summary Specifications) are included in the CC evaluation scope. Other product features and functions not included in the scope of this ST are deemed unevaluated and non-interfering. These features and functions include the following:

- Data reduction technology

- Non-Disruptive Expansion and High Availability
- Snapshots, Backup & Disaster Recovery including asynchronous replication, ActiveDR, ActiveCluster, and SafeMode
- Data at rest encryption
- Graphical User Interface
- REST API
- Remote Assist

# 2 Conformance Claims

This section identifies the TOE conformance claims, conformance rationale, and relevant Technical Decisions (TDs).

## 2.1 CC Conformance Claims

The TOE is conformant to the following:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017 (Extended)
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision5, April 2017 (Conformant)

## 2.2 Protection Profile Conformance

This ST claims exact conformance to the following:

- Collaborative Protection Profile for Network Devices, Version 3.0e, December 6, 2023 [PP-ND]
- Functional Package for Secure Shell (SSH) Version 1.0, May 13, 2021 [PKG-SSH].

## 2.3 Conformance Rationale

This ST provides exact conformance to the items listed in the previous section. The security problem definition, security objectives, and security requirements in this ST are all taken from the NDcPP v3.0e [PP-ND], performing only the operations defined there.

### 2.3.1 Technical Decisions

All NIAP Technical Decisions (TDs) issued to date and applicable to NDcPP v3.0e [PP-ND] and PKG_SSH v1.0 [PKG-SSH] have been considered. Table 6 identifies all applicable TDs.

**Table 6 - Relevant Technical Decisions**

| Technical Decision | Applicable PP | Applicable (Y/N) | Exclusion Rationale (if applicable) |
|---|---|---|---|
| TD0923.Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2 | NDcPP v3.0e | Yes | |
| TD0921. NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment | NDcPP v3.0e | Yes | |
| TD0900. Clarification to Local Administrator Access in FIA_UIA_EXT.1.3 | NDcPP v3.0e | Yes | |
| TD0899. Correction of Renegotiation Test for TLS 1.2 | NDcPP v3.0e | Yes | |
| TD0886: NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1 | NDcPP v3.0e | Yes | |
| TD0880: NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1 | NDcPP v3.0e | Yes | |
| TD0879: NIT Decision: Correction of Chapter Headings in CPP_ND_V3.0E | NDcPP v3.0e | Yes | |

| Technical Decision | Applicable PP | Applicable (Y/N) | Exclusion Rationale (if applicable) |
|---|---|---|---|
| TD0868: NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8 | NDcPP v3.0e | No | FCS_IPSEC_EXT.1 is not claimed by the TOE. |
| TD0836: NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1 | NDcPP v3.0e | Yes | |
| TD0909: Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0 | PKG_SSH v1.0 | Yes | |
| TD0777: Clarification to Selections for Auditable Events for FCS_SSH_EXT.1 | PKG_SSH v1.0 | Yes | |
| TD0732: FCS_SSHS_EXT.1.3 Test 2 Update | PKG_SSH v1.0 | Yes | |
| TD0695: Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package. | PKG_SSH v1.0 | Yes | |
| TD0682: Addressing Ambiguity in FCS_SSHS_EXT.1 Tests | PKG_SSH v1.0 | Yes | |

# 3 Security Problem Definition

The security problem definition has been taken directly from the NDcPP v3.0e and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any Organizational Security Policies (OSPs) that the TOE is expected to enforce.

## 3.1 Threats

The threats included in Table 7 are drawn directly from the NDcPP v3.0e specified in Section 2.2.

**Table 7 - Threats**

| ID | Threat |
|---|---|
| T.UNAUTHORIZED_ADMINISTRATOR_ACCESS | Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides. |
| T.WEAK_CRYPTOGRAPHY | Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort. |
| T.UNTRUSTED_COMMUNICATION_CHANNELS | Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself. |
| T.WEAK_AUTHENTICATION_ENDPOINTS | Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints, e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised. |

| ID | Threat |
|---|---|
| T.UPDATE_COMPROMISE | Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration. |
| T.UNDETECTED_ACTIVITY | Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised. |
| T.SECURITY_FUNCTIONALITY_COMPROMISE | Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker. Threat agents may also be able to take advantage of weak administrative passwords to gain privileged access to the device. |
| T.SECURITY_FUNCTIONALITY_FAILURE | An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device. |

## 3.2 Assumptions

The assumptions included in Table 8 are drawn directly from the NDcPP v3.0e .

**Table 8 - Assumptions**

| ID | Assumption |
|---|---|
| A.PHYSICAL_PROTECTION | The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs. |
| A.LIMITED_FUNCTIONALITY | The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality). |
| A.NO_THRU_TRAFFIC_PROTECTION | A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall). |
| A.TRUSTED_ADMINISTRATOR | The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.<br><br>For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification). |

| ID | Assumption |
|---|---|
| A.REGULAR_UPDATES | The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities. |
| A.ADMIN_CREDENTIALS_SECURE | The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside. |
| A.RESIDUAL_INFORMATION | The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. |

## 3.3 Organizational Security Policies

The OSPs included in Table 9 are drawn directly from the NDcPP v3.0e .

**Table 9 - OSPs**

| ID | OSP |
|---|---|
| P.ACCESS_BANNER | The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE. |

# 4 Security Objectives

The security objectives have been taken directly from the NDcPP v3.0e and are reproduced here for the convenience of the reader.

## 4.1 Security Objectives for the Operational Environment

Security objectives for the operational environment assist the TOE in correctly providing its security functionality. These objectives, which are found in the table below, track with the assumptions about the TOE operational environment.

**Table 10 - Security Objectives for the Operational Environment**

| ID | 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. |
| 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_ADMN | Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.<br><br>For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted. |
| OE.UPDATES | The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities. |
| OE.ADMIN_CREDENTIALS_SECURE | The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside. |
| OE.RESIDUAL_INFORMATION | The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. |

# 5 Security Requirements

This section identifies the Security Functional Requirements (SFRs) for the TOE. The SFRs included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, September 2017, and all international interpretations.

**Table 11 - SFRs**

| Requirement | Description | Location |
|---|---|---|
| FAU_GEN.1 | Audit Data Generation | NDcPP v3.0e |
| FAU_GEN.2 | User Identity Association | NDcPP v3.0e |
| FAU_STG.1 | Protected Audit Trail Storage | NDcPP v3.0e |
| FAU_STG_EXT.1 | Protected Audit Event Storage | NDcPP v3.0e |
| FCS_CKM.1 | Cryptographic Key Generation | NDcPP v3.0e |
| FCS_CKM.2 | Cryptographic Key Establishment | NDcPP v3.0e |
| FCS_CKM.4 | Cryptographic Key Destruction | NDcPP v3.0e |
| FCS_COP.1/DataEncryption | Cryptographic Operation (AES Data Encryption/Decryption) | NDcPP v3.0e |
| FCS_COP.1/SigGen | Cryptographic Operation (Signature Generation and Verification) | NDcPP v3.0e |
| FCS_COP.1/Hash | Cryptographic Operation (Hash Algorithm) | NDcPP v3.0e |
| FCS_COP.1/KeyedHash | Cryptographic Operation (Keyed Hash Algorithm) | NDcPP v3.0e |
| FCS_RBG_EXT.1 | Random Bit Generation | NDcPP v3.0e |
| FCS_NTP_EXT.1 | NTP Protocol | NDcPP v3.0e |
| FCS_SSH_EXT.1 | SSH Protocol | PKG_SSH v1.0 |
| FCS_SSHS_EXT.1 | SSH Protocol - Server | PKG_SSH v1.0 |
| FCS_TLSC_EXT.1 | TLS Client Protocol without Mutual Authentication | NDcPP v3.0e |
| FIA_AFL.1 | Authentication Failure Management | NDcPP v3.0e |
| FIA_PMG_EXT.1 | Password Management | NDcPP v3.0e |
| FIA_UIA_EXT.1 | User Identification and Authentication | NDcPP v3.0e |
| FIA_UAU.7 | Protected Authentication Feedback | NDcPP v3.0e |
| FIA_X509_EXT.1/Rev | X.509 Certificate Validation | NDcPP v3.0e |
| FIA_X509_EXT.2 | X.509 Certificate Authentication | NDcPP v3.0e |
| FIA_X509_EXT.3 | X.509 Certificate Requests | NDcPP v3.0e |
| FMT_MOF.1/ManualUpdate | Management of Security Functions Behaviour | NDcPP v3.0e |
| FMT_MTD.1/CoreData | Management of TSF Data | NDcPP v3.0e |
| FMT_MTD.1/CryptoKeys | Management of TSF Data | NDcPP v3.0e |
| FMT_SMF.1 | Specification of Management Functions | NDcPP v3.0e |
| FMT_SMR.2 | Restrictions on security roles | NDcPP v3.0e |
| FPT_SKP_EXT.1 | Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) | NDcPP v3.0e |
| FPT_APW_EXT.1 | Protection of Administrator Passwords | NDcPP v3.0e |
| FPT_TST_EXT.1 | TSF Testing | NDcPP v3.0e |
| FPT_STM_EXT.1 | Reliable Time Stamps | NDcPP v3.0e |
| FPT_TUD_EXT.1 | Trusted Update | NDcPP v3.0e |
| FTA_SSL.3 | TSF-initiated Termination | NDcPP v3.0e |
| FTA_SSL.4 | User-initiated Termination | NDcPP v3.0e |
| FTA_SSL_EXT.1 | TSF-initiated Session Locking | NDcPP v3.0e |
| FTA_TAB.1 | Default TOE Access Banner | NDcPP v3.0e |
| FTP_ITC.1 | Inter-TSF Trusted Channel | NDcPP v3.0e |
| FTP_TRP.1/Admin | Trusted Path | NDcPP v3.0e |

## 5.1 Conventions

The CC allows the following types of operations to be performed on the functional requirements: assignments, selections, refinements, and iterations. The following font conventions are used within this document to identify operations defined by CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending the iteration identifier after a slash, e.g., /SigGen.
- Where operations were completed in the PPs, the formatting used in the PPs has been retained.
- Extended SFRs are identified by the addition of "EXT" after the requirement name.

## 5.2 Security Functional Requirements

This section includes the security functional requirements for this ST.

### 5.2.1 Security Audit (FAU)

#### 5.2.1.1 FAU_GEN.1 Audit Data Generation

**FAU_GEN.1.1**

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) *All* administrative *actions comprising:*
   - *Administrative login and logout (name of Administrator account shall be logged if individual 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 Administrator account shall be logged)];*
d) *Specifically defined auditable events listed in* ***Table 12***.

**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 ~~(if applicable)~~, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of* **Table 12**.

**Table 12 – Security Functional Requirements and Auditable Events**

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FAU_GEN.1 | None | None |
| FAU_GEN.2 | None | None |
| FAU_STG.1 (OP) | None | None |
| FAU_STG_EXT.1 (TD0923) | Configuration of local audit settings. | Identity of account making changes to the audit configuration. |
| FCS_CKM.1 | None | None |
| FCS_CKM.2 | None | None |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FCS_CKM.4 | None | None |
| FCS_COP.1/DataEncryption | None | None |
| FCS_COP.1/SigGen | None | None |
| FCS_COP.1/Hash | None | None |
| FCS_COP.1/KeyedHash | None | None |
| FCS_RBG_EXT.1 | None | None |
| FCS_NTP_EXT.1 (SB) | ● Configuration of a new time server<br>● Removal of configured time server | Identity if new/removed time server |
| FCS_SSH_EXT.1 (SSH) | ● Failure to establish a SSH session<br>● Establishment of SSH Connection<br>● Termination of SSH connection session | [Reason for failure and Non-TOE endpoint of attempted connection (IP Address) |
| | [*None*] | [*None*] |
| FCS_SSHS_EXT.1 (SSH-SB) | None | None |
| FCS_TLSC_EXT.1 (SB) | Failure to establish a TLS Session | Reason for failure |
| FIA_AFL.1 (SB) | Unsuccessful login attempts limit is met or exceeded | Origin of the attempt (e.g., IP address) |
| FIA_PMG_EXT.1 (SB) | None | None |
| FIA_UIA_EXT.1 | All use of identification and authentication mechanism | Origin of the attempt (e.g., IP address) |
| FIA_UAU.7 (SB) | None | None |
| FIA_X509_EXT.1/Rev (SB) | ● Unsuccessful attempt to validate a certificate<br>● Any addition, replacement or removal of trust anchors in the TOE's trust store | ● Reason for failure of certificate validation<br>● Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store |
| FIA_X509_EXT.2 (SB) | None | None |
| FIA_X509_EXT.3 (SB) | None | None |
| FMT_MOF.1/ManualUpdate | Any attempt to initiate a manual update | None |
| FMT_MTD.1/CoreData | None | None |
| FMT_MTD.1/CryptoKeys (SB) | None | None |
| FMT_SMF.1 | All management activities of TSF data | None |
| FMT_SMR.2 | None | None |
| FPT_SKP_EXT.1 | None | None |
| FPT_APW_EXT.1 (SB) | None | None |
| FPT_TST_EXT.1 | None. | None. |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FPT_STM_EXT.1 | Discontinuous changes to time - either Administrator actuated or changed via an automated process<br><br>(Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1) | For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address). |
| FPT_TUD_EXT.1 | Initiation of update; result of the update attempt (success or failure) | 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_SSL_EXT.1 (if "terminate the session" is selected) (SB) | The termination of a local session by the session locking mechanism | None |
| FTA_TAB.1 | None | None |
| FTP_ITC.1 | ● Initiation of the trusted channel<br>● Termination of the trusted channel<br>● Failure of the trusted channel functions | ● None<br>● None<br>● Reason for failure |
| FTP_TRP.1/Admin | ● Initiation of the trusted path<br>● Termination of the trusted path.<br>● Failure of the trusted path functions. | ● None<br>● None<br>● Reason for failure |

**Application Note:** "(SB)", "(OP)", and "(SSH)" indications in Table 12 have the following meanings:
- (SB) - Selection Based SFR, from Section B of the [PP-ND]
- (OP) - Optional SFR, from Section A of the [PP-ND].
- (SSH) - Mandatory SFR, from Section 3 of [PKG-SSH].
- (SSH-SB) - Selection Based SFR, from Section B of [PKG-SSH].

**Application Note:** "(TD0923)" indication in Table 12 has the following meanings:
- (1) - Auditable event for FAU_STG_EXT.1 is not supported as the TOE does not support configuration of local audit settings.

**Application Note:** This table has been updated as per TD0777, TD0923.

#### 5.2.1.2 FAU_GEN.2 User Identity Association

**FAU_GEN.2.1**
For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

#### 5.2.1.3 FAU_STG.1 Protected Audit Trail Storage

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

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

#### 5.2.1.4 FAU_STG_EXT.1 Protected Audit Event Storage

**FAU_STG_EXT.1.1**
The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according 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 maintain a [*database*] of audit records in the event that an interruption of communication with the remote audit server occurs.

**FAU_STG_EXT.1.4**
The TSF shall be able to store [*persistent*] audit records locally with a minimum storage size of [*up to 1000 entries for each of the three audit log categories*].

**FAU_STG_EXT.1.5**
The TSF shall [*overwrite previous audit records according to the following rule:* [*overwrite oldest audit record ('eat tail')*]] when the local storage space for audit data is full.

**FAU_STG_EXT.1.6**
The TSF shall provide the following mechanisms for administrative access to locally stored audit records [*ability to view locally*].

### 5.2.2 Cryptographic Support (FCS)

#### 5.2.2.1 FCS_CKM.1 Cryptographic Key Generation

**FCS_CKM.1.1**
The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- *RSA schemes using cryptographic key sizes of [2048 bits] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;*

- *ECC schemes using elliptic curves [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4, or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.*

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards]~~.

**Application Note:** TD0921 applied

#### 5.2.2.2 FCS_CKM.2 Cryptographic Key Establishment

**FCS_CKM.2.1**
The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [

- *RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2";*
- *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";*

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

#### 5.2.2.3 FCS_CKM.4 Cryptographic Key Destruction

**FCS_CKM.4.1**
The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- *For plaintext keys in volatile storage, the destruction shall be executed by a [single overwrite consisting of [zeroes]];*
- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [*
  - *logically addresses the storage location of the key and performs a [[3]-pass] overwrite consisting of [[a pseudorandom pattern]]*];

that meets the following: *No Standard.*

#### 5.2.2.4 FCS_COP.1/DataEncryption Cryptographic Operations (AES Data Encryption/Decryption)

**FCS_COP.1.1/DataEncryption**
The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in* [*CBC, 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 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

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

- *RSA Digital Signature Algorithm*

- *Elliptic Curve Digital Signature Algorithm*

]
and cryptographic key sizes [
- *For RSA: [2048 bits],*
- *For ECDSA: [256, 384, 512 bits]*

]
that meet the following: [
- *For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5 using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4, using PKCS #1 v2.2 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 implementing [P-256, P-384, P-521] curves that meet the following : FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST Recommended" curves; or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.*

].

**Application Note:** TD0921 applied

#### 5.2.2.6 FCS_COP.1/Hash Cryptographic Operations (Hash Algorithm)

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

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

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

#### 5.2.2.8 FCS_RBG_EXT.1 Random Bit Generation

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

**FCS_RBG_EXT.1.2**
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[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.

#### 5.2.2.9 FCS_NTP_EXT.1 NTP Protocol

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

**FCS_NTP_EXT.1.2**
The TSF shall update its system time using [
- Authentication using [*SHA256*] as the message digest algorithm(s);

].

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

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

#### 5.2.2.10 FCS_SSH_EXT.1 SSH Protocol

**FCS_SSH_EXT.1.1**
The TOE shall implement SSH acting as a [*server*]in accordance with: RFCs 4251, 4252, 4253, 4254, [*4344,* 5647, *5656, 6668, 8308, 8332*] and [no other standard].

**FCS_SSH_EXT.1.2**
The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [
- *"password" (RFC 4252),*
- *"publickey" (RFC 4252): [*
  - *rsa-sha2-256 (RFC 8332),*
  - *rsa-sha2-512 (RFC 8332),*

  *]*

] and no other methods.

**FCS_SSH_EXT.1.3**
The TSF shall ensure that, as described in RFC 4253, packets greater than [*262,126*] bytes in an SSH transport connection are dropped.

**FCS_SSH_EXT.1.4**
The TSF shall protect data in transit from unauthorized disclosure using the following mechanisms: [
- *aes128-ctr (RFC 4344),*
- *aes256-ctr (RFC 4344),*
- *aes128-cbc (RFC 4253),*
- *aes256-cbc (RFC 4253),*
- *aes128-gcm@openssh.com (RFC 5647),*
- *aes256-gcm@openssh.com (RFC 5647)*

] and no other mechanisms.

**FCS_SSH_EXT.1.5**
The TSF shall protect data in transit from modification, deletion, and insertion using: [
- *hmac-sha2-256 (RFC 6668),*
- *hmac-sha2-512 (RFC 6668),*

- *implicit*

] and no other mechanisms.

**FCS_SSH_EXT.1.6**
The TSF shall establish a shared secret with its peer using: [

- *ecdh-sha2-nistp256 (RFC 5656),*
- *ecdh-sha2-nistp384 (RFC 5656),*
- *ecdh-sha2-nistp521 (RFC 5656),*

] and no other mechanisms.

**FCS_SSH_EXT.1.7**
The TSF shall use *SSH KDF* as defined in [

- *RFC 5656 (Section 4)*

] to derive the following cryptographic keys from a shared secret: *session keys*.

**FCS_SSH_EXT.1.8**
The TSF shall ensure that [

- *a rekey of the session keys,*

] occurs when any of the following thresholds are met:

- one hour connection time
- no more than one gigabyte of transmitted data, or
- no more than one gigabyte of received data.

#### 5.2.2.11 FCS_SSHS_EXT.1 SSH Protocol - Server

**FCS_SSHS_EXT.1.1**
The TSF shall authenticate itself to its peer (SSH Client) using: [

- *rsa-sha2-256 (RFC 8332),*
- *rsa-sha2-512 (RFC 8332),*

].

#### 5.2.2.12 FCS_TLSC_EXT.1 TLS Client Protocol without Mutual Authentication

**FCS_TLSC_EXT.1.1**
The TSF shall implement [ *TLS 1.2 (RFC 5246)*] supporting the following ciphersuites:
[

- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_ECDHE_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, IPv4 address in SAN, and no other attribute types*].

**FCS_TLSC_EXT.1.3**
The TSF shall not establish a trusted channel if the server certificate is invalid [

- *Without any administrator override mechanism.*

].

**FCS_TLSC_EXT.1.4**
The TSF shall [*present the Supported Groups Extension with the following curves/groups: [secp256r1, secp384r1, secp521r1] and no other curves/groups*] in the Client Hello.

**FCS_TLSC_EXT.1.5**
The TSF shall [

- *present the signature_algorithms extension with support for the following algorithms: [*
  - *rsa_pkcs1 with sha256(0x0401),*
  - *rsa_pkcs1with sha384(0x0501),*
  - *rsa_pkcs1 with sha512(0x0601),*
  - *ecdsa_secp256r1 with sha256(0x0403),*
  - *ecdsa_secp384r1 with sha384(0x0503),*
  - *ecdsa_secp521r1 with sha512(0x0603),*
  - *] and no other algorithms;*

].

**FCS_TLSC_EXT.1.6**
The TSF [*does not provide*] the ability to configure the list of supported ciphersuites as defined in FCS_TLSC_EXT.1.1.

**FCS_TLSC_EXT.1.7**
The TSF shall prohibit the use of the following extensions:

- Early data extension
- Post-handshake client authentication according to RFC 8446, Section 4.2.6.

**FCS_TLSC_EXT.1.8**
The TSF shall [*not use PSKs*].

**FCS_TLSC_EXT.1.9**
The TSF shall [reject [*TLS 1.2*] renegotiation attempts].

### 5.2.3 Identification and Authentication (FIA)

#### 5.2.3.1 FIA_AFL.1 Authentication Failure Management

**FIA_AFL.1.1**
The TSF shall detect when an Administrator configurable positive integer within [*1 to 100*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

**FIA_AFL.1.2**
When the defined number of unsuccessful authentication attempts has been met, the TSF shall [*prevent the offending 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 FIA_PMG_EXT.1 Password Management

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

*a)* Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [*"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [ASCII hexadecimal codes 0x20-0x2F, 0x3A-0x40, 0x5B-0x60, 0x7B-0x7E]*]

*b)* Minimum password length shall be configurable to between [*15*] and [*100*] characters.

#### 5.2.3.3 FIA_UIA_EXT.1 User Identification and Authentication

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

- Display the warning banner in accordance with FTA_TAB.1;
- [*[Respond to ICMP Echo Request, Respond to ARP requests with ARP replies, Make DNS Requests]*].

**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.

**FIA_UIA_EXT.1.3**
The TSF shall provide the following remote authentication mechanisms [*SSH password, SSH public key*] and [*no other mechanism*]. The TSF shall provide the following local authentication mechanisms [*password-based*].

**Application Note:** TD0900 applied

**FIA_UIA_EXT.1.4**
The TSF shall authenticate any administrative user's claimed identity according to each authentication mechanism specified in FIA_UIA_EXT.1.3.

#### 5.2.3.4 FIA_UAU.7 Protected Authentication Feedback

**FIA_UAU.7.1**
The TSF shall provide only *obscured feedback* to the **administrative** user while the authentication is in progress **at the local console.**

#### 5.2.3.5 FIA_X509_EXT.1/Rev X.509 Certificate Validation

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

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

#### 5.2.3.6 FIA_X509_EXT.2 X.509 Certificate Authentication

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

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

#### 5.2.3.7 FIA_X509_EXT.3 X.509 Certificate Requests

**FIA_X509_EXT.3.1**
The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [*Common Name, 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 FMT_MOF.1/ManualUpdate Management of Security Functions Behavior

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

#### 5.2.4.2 FMT_MTD.1/CoreData Management of TSF Data

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

#### 5.2.4.3 FMT_MTD.1/CryptoKeys Management of TSF Data

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

#### 5.2.4.4 FMT_SMF.1 Specification of Management Functions

**FMT_SMF.1.1**
The TSF shall be capable of performing the following management functions:

- *Ability to administer the TOE remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the remote session inactivity time before session termination;*
- *Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;*
- [
    - *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*
    - *Ability to manage the cryptographic keys;*
    - *Ability to configure the cryptographic functionality;*
    - *Ability to re-enable an Administrator account;*
    - *Ability to configure NTP;*
    - *Ability to configure the reference identifier for the peer;*
    - *Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;*
    - *Ability to generate Certificate Signing Request (CSR) and process CA certificate response;*
    - *Ability to administer the TOE locally;*
    - *Ability to configure the local session inactivity time before session termination or locking;*
    - *Ability to configure the authentication failure parameters for FIA_AFL.1;*
    - *Ability to manage the trusted public keys database;*
    - *No other capabilities*

].

**Application Note:** TD0836 applied

#### 5.2.4.5 FMT_SMR.2 Restrictions on Security Roles

**FMT_SMR.2.1**
The TSF shall maintain the roles:

- *Security Administrator*

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

**FMT_SMR.2.3**
The TSF shall ensure that the conditions

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

are satisfied.

### 5.2.5 Protection of the TSF (FPT)

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

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

#### 5.2.5.2 FTP_APW_EXT.1 Protection of Administrator Passwords

**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.

#### 5.2.5.3 FPT_TST_EXT.1 TSF Testing

**FPT_TST_EXT.1.1**
The TSF shall run a suite of the following self-tests:

- *During initial start-up (on power on) to verify the integrity of the TOE firmware and software;*
- *Prior to providing any cryptographic service and [at no other time] to verify correct operation of cryptographic implementation necessary to fulfil the TSF;*
- *[no other] self-tests [are run].*

to demonstrate the correct operation of the TSF.

**Application Note:** This SFR has been updated as per TD0836.

**FPT_TST_EXT.1.2**
The TSF shall respond to [*all failures*] by [*halting TOE start-up with local console displayed error message until the TOE is power-cycled*].

#### 5.2.5.4 FPT_TUD_EXT.1 Trusted Update

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

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

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

#### 5.2.5.5 FPT_STM_EXT.1 Reliable Time Stamps

**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 [*synchronise time with an NTP server*].

### 5.2.6 TOE Access (FTA)

#### 5.2.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

**FTA_SSL_EXT.1.1**
The TSF shall, for local interactive sessions, [

- *terminate the session*

]
after a Security Administrator-specified time period of inactivity

#### 5.2.6.2 FTA_SSL.3 TSF-initiated Termination

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

#### 5.2.6.3 FTA_SSL.4 User-initiated Termination

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

#### 5.2.6.4 FTA_TAB.1 Default TOE Access Banners

**FTA_TAB.1.1**
Before establishing **an administrative user** session the TSF shall display **a Security Administrator-specified** advisory **notice and consent** warning message regarding ~~unauthorized~~ use of the TOE.

### 5.2.7 Trusted Path/Channels (FTP)

#### 5.2.7.1 FTP_ITC.1 Inter-TSF Trusted Channel

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

**FTP_ITC.1.2**
The TSF shall permit [*the TSF*] to initiate communication via the trusted channel.

**FTP_ITC.1.3**
The TSF shall initiate communication via the trusted channel for *[audit transfer]*.

#### 5.2.7.2 FTP_TRP.1/Admin Trusted Path

**FTP_TRP.1.1/Admin**
The TSF shall **be capable of using [*SSH*] to** provide a communication path between itself and **authorized** remote **Administrators** ~~users~~ 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** ~~users~~ to initiate communication via the trusted path.

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

## 5.3 TOE SFR Dependencies Rationale for SFRs

NDcPP v3.0e [PP-ND] and PKG_SSH v1.0 [PKG-SSH] contain all the requirements claimed in this ST. As such, dependencies are not applicable since the NDcPP v3.0e [PP-ND] and PKG_SSH v1.0 [PKG-SSH] have been approved.

## 5.4 Security Assurance Requirements

The security assurance requirements for the TOE are included by reference to [PP-ND] and [PKG-SSH].

**Table 13 - Security Assurance Requirements**

| Assurance Class | Assurance Components | Component Description |
|---|---|---|
| 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 |
| Development | ADV_FSP.1 | Basic functional specification |
| Guidance Documents | AGD_OPE.1 | Operational user guidance |
| | AGD_PRE.1 | Preparative user guidance |
| Life Cycle Support | ALC_CMC.1 | Labelling of the TOE |
| | ALC_CMS.1 | TOE CM coverage |
| Tests | ATE_IND.1 | Independent testing – conformance |
| Vulnerability Assessment | AVA_VAN.1 | Vulnerability survey |

Consequently, the evaluation activities specified in the following Supporting Documents apply to the TOE evaluation:

- Evaluation Activities for Network Device cPP, December-2023, Version 3.0e
- Functional Package for Secure Shell (SSH), May 13, 2021, Version 1.0

# 6 TOE Summary Specifications

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

**Table 14 – TOE Summary Specification SFR Description**

| Requirement | TSS Description |
|---|---|
| FAU_GEN.1 | The TOE utilizes the syslog system library built into the underlying Linux kernel of the TOE to generate local audit records. The TOE uses a custom database for audit log storage, described below in FAU_STG_EXT.1 TSS. It is the responsibility of each calling application (such as SSH or OpenSSL) to call the syslog function, which forwards the audit log messages to the appropriate destination (local database, remote syslog server). Within the TSF, a second, logically distinct call to syslog is made when generating audit logs destined for the external audit log server, ensuring only security-relevant logs reach the audit log server. The TSF includes functionality that uses the syslog system library to generate audit records specifically for the audit requirements specified in Table 12: Auditable Events, as well as start-up and shut-down of the audit functions.<br><br>The syslog daemon will automatically record the date and time (accurate to the second) for each event.<br><br>The TSF categorizes logs into three categories (or type of event), user session, configuration changes, and alert records, based on the function call made to the syslog daemon. The storage location and functionality of these audit records are described in FAU_STG_EXT.1 TSS below.<br><br>Authenticated administrators can view the audit records.<br><br>For all audit records, each audit record will contain the subject identity and outcome of the event within the audit log message. The format of audit log messages is described in operational guidance [AGD].<br><br>The CLI allows restricted access only to custom Pure Storage binaries, each of which contain calls to syslog when necessary and capture the currently logged-in user identity.<br><br>All audit log messages created by the TOE that are relevant to the functions described in this document are described in the guidance documentation [AGD].<br><br>The TOE generates audit records for all of the events defined for FAU_GEN.1.1 listed in:<br>• NDcPP v3.0e Section 6.3.1.1 and SFR-specific auditable events as defined in NDcPP v3.0e Table 2, Table 3, and Table 4 for claimed SFRs.<br>• PKG_SSH v1.0 Section 3.1 and B.1 auditable events as defined in PKG_SSH v1.0 Table 1 and Table 2.<br><br>The ‘Start up and shutdown of the audit function’ audit record is captured as start up and shutdown messages for the TOE itself, since logging may not be started or stopped independently of powering on and powering off the TOE. |

| Requirement | TSS Description |
|---|---|
| | Only successfully identified and authenticated administrators can view/read, logs on the TOE.<br><br>The TSF generates an audit record anytime a persistent cryptographic key is created, modified or destroyed. The audit records identify the name of the cryptographic key in question. The name of the cryptographic key is set by the administrator at the time of generation of the key. |
| FAU_GEN.2 | For each audit event generated on the TOE, at minimum, the TSF associates the audit event with the identity of the user that caused the event, and contains date-and-time stamp, type, subject identity, and outcome (success or failure) of the event, and any SFR-specific additional audit record contents required as described in column three of Table 12. |
| FAU_STG.1 | Audit records are protected from unauthorized access by restrictive CLI (local console, SSH console), which only allows authorized administrators to edit audit-related settings. The TSF protects the locally stored audit records in the audit trail from unauthorized deletion via the user authentication and access control mechanism of the TOE. Security Administrators must be successfully authenticated to the TOE to view locally stored audit records. |
| FAU_STG_EXT.1 | The TSF secures the transmission of audit records to the remote audit server using syslog over TLS with syslog-ng and OpenSSL.<br><br>As described in FAU_GEN.1 TSS above, the TSF contains a custom database that contains user-specific configurations and audit log entries. The local audit log server maintains a database table of up to 1000 entries for each of the three audit log categories: user session, configuration changes, and alert records. The database is persistently stored on the locally connected SAS drives where capacity is managed by the operational environment but will typically have 100-1000x the capacity for the audit logs required (approximately 170 GB). When the local storage space for audit records is full, the TSF deletes the oldest audit log and then records the newest audit log entry (eat-tail method).<br><br>All audit records generated on the TOE are sent to a remote audit server over the TLS protected Trusted Channel. The audit records that are stored locally on the standalone TOE and those sent to the remote audit server are identical in content and format. Locally generated audit records are sent to the remote audit server as soon as they are generated. If the Trusted Channel is not operational, then audit records will not be sent to the remote audit server; however, they will still be locally stored. The TSF does not queue up audit records that were not sent to the remote audit server for transmitting upon re-establishment of the Trusted Channel.<br><br>Only authenticated administrators can access the TOE's internal audit log database through the `pureaudit`, `purealert`, and `puremessage` CLI commands, which provide only viewing access. The TOE does not implement any CLI commands or interfaces to modify or delete records stored in the database. The underlying database querying functions do not accept commands to modify or delete audit records. |

| Requirement | TSS Description |
|---|---|
| | The TOE does not support configuration of local audit settings. Accordingly, the audit event for FAU_STG_EXT.1 indicated in Table 12 is not supported by the TOE (TD0923). |
| FCS_CKM.1 | The TOE fulfills all of the NIST SP 800-56A requirements for supported algorithms without extensions. The TOE does not perform any operations marked as "shall not" or "should not" and performs all operations marked as "shall" or "should."<br><br>The TOE utilizes OpenSSL for the generation of asymmetric keys. Asymmetric keys are generated for SSH and TLS authentication and key-exchange<br><br>The following asymmetric keys and key sizes are generated by the TOE:<br>● For SSH Server TSF:<br>○ Authentication:<br>▪ rsa-sha2 key pairs (2048 bits)<br>▪ ECDSA key pairs (256/384/521 bits)<br>○ Key-Exchange:<br>▪ ECDH key-pairs<br>● ecdh-sha2-nistp256 (P-256)<br>● ecdh-sha2-nistp384 (P-384)<br>● ecdh-sha2-nistp521 (P-521)<br>● For TLS Client TSF:<br>○ RSA key pairs (2048 bits)<br>○ ECDSA key-pairs (256 bits) |
| FCS_CKM.2 | The TOE utilizes cryptographic key-establishment schemes when negotiating an SSH Server Trusted Path and a TLS Trusted Channel. The following list provides the key-establishment schemes utilized and the purpose of their use:<br>● RSA-based key establishment scheme that meets RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1.<br>○ When RSA-key-transport-based ciphersuites are negotiated/used during TLS session negotiation to the TLS sessions to a remote audit server.<br>▪ TOE acts as a session initiator<br><br>● 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"<br>○ When ECDH-key-transport-based key exchange methods negotiated/used during SSH session negotiation to the TOE remote access SSH server.<br>▪ TOE acts as a session recipient.<br>○ When ECDHE-key-transport-based key exchange methods negotiated/used during TLS session negotiation from the TOE remote Syslog server.<br>▪ TOE acts as a session initiator. |

| Requirement | TSS Description |
|---|---|
| FCS_CKM.4 | The table below describes each of the secret keys, private keys, and CSPs used to generate keys: |

**Table 15: Cryptographic CSPs**

| CSP Name & Library | Description | Storage |
|---|---|---|
| RSA SGK - OpenSSL | RSA (2048 bits) Signature Generation Key | Volatile memory (RAM) |
| ECDSA SGK OpenSSL | ECDSA (256, 384, 512 bits) Signature Generation Key | Volatile memory (RAM) |
| AES EDK - OpenSSL and OpenSSH | AES (128/256) Encrypt/Decrypt Key | Volatile memory (RAM) |
| HMAC Key - OpenSSL and OpenSSH | Keyed hash key (256, 384, 512 bits) | Volatile memory (RAM) |
| RSA Private - OpenSSH | RSA private key agreement key | Volatile memory (RAM) |
| ECDH Private - OpenSSL and OpenSSH | ECDH (Diffie-Hellman) private key agreement key | Volatile memory (RAM) |
| RNG CSPs - OpenSSL | Entropy input (361 bits) SP800-90A based RNG. Used to generate keys listed above and the SSH private key. | Volatile memory (RAM) |
| Server Private Keys - OpenSSH | Private RSA and ECDSA keys for OpenSSH authentication | Volatile memory (RAM), Local Filesystem (SSD) |
| Server Private Keys - TLS | Private RSA and ECDSA keys for TLS authentication, syslog-ng | Volatile memory (RAM) |

The table below describes the public keys used as part of the cryptographic processes within the TSF:

**Table 16: Cryptographic Public Keys**

| Public Key Name & Library | Description | Storage |
|---|---|---|

| Requirement | TSS Description | | |
|---|---|---|---|
| | RSA SVK – OpenSSL | RSA (2048 bits) Signature Verification Key | Volatile memory (RAM) |
| | ECDSA SVK – OpenSSL | ECDSA (256, 384, 512 bits) Signature Verification Key | Volatile memory (RAM) |
| | RSA KEK – OpenSSL | RSA (2048 bits) Key Encryption (public key transport) Key | Volatile memory (RAM) |
| | ECDH Public - OpenSSL and OpenSSH | ECDH (P-256, P-384, and P-521) public key agreement key | Volatile memory (RAM) |
| | Server Public Keys – OpenSSH | Public RSA and ECDSA keys for OpenSSH authentication | Volatile memory (RAM), Local Filesystem (SSD) |
| | The TSF zeroizes volatile secret and private keys when power is removed. As the power is removed from the volatile memory, the RAM loses its charge, and thus all data is lost after a short amount of time.<br><br>Persistent cryptographic keys are instantiated by the administrator in the following scenarios:<br>• when an SSH host-key is needed for the SSH Server TSF,<br>• when generating a CSR or importing an externally generated key into the TSF's trust store.<br><br>These persistent cryptographic keys are stored in the underlying filesystem of the TOE (non-volatile storage).<br><br>Persistent keys in non-volatile storage can be securely erased by the Security Administrator performing a secure erase procedure using the `reset_apartment` command that performs a 3-pass overwrite using a pseudorandom pattern followed by a reboot of the TOE.<br><br>Temporary session keys for TLS and SSH, or persistent keys loaded into memory for use to perform cryptographic operations, are plaintext keys. The session keys are generated as SSH and TLS sessions are negotiated. Both the ephemerally generated session keys and the memory-loaded persistent keys are stored only in RAM (volatile memory) and are zeroized when the associated process that generated them is terminated or rekeying on a particular trusted channel/path session has occurred. Zeroization of these keys occurs by a single overwrite of the key-data, consisting of zeroes. | | |
| FCS_COP.1/DataEncryption; | The TOE performs encryption/decryption using AES in CBC, CTR and GCM modes, using key sizes of 128 or 256 bits, depending on which TLS and/or SSH ciphers are negotiated. This is enforced by the OpenSSL cryptographic module and is not configurable by the security administrator. This functionality is performed as | | |

| Requirement | TSS Description |
|---|---|
| | specified for AES in ISO 18033-3, for CBC as specified in ISO 10116, for CTR as specified in ISO 10116, and for GCM as specified in ISO 19772. |
| FCS_COP.1/SigGen; | The TOE provides RSA and ECDSA digital signatures (signature generation and signature verification) for ECDSA-based (SSH and TLS) and RSA-based (SSH and TLS) cryptographic functions performed in TLS Client and SSH Server TSFs. The implementation of RSA key generation for these security functions conforms to FIPS Pub 186-4 Appendix B.3. RSA key size of 2048-bits is supported for TLS (client). The implementation of ECDSA key generation for these security functions conform to FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4 with key sizes of P-256, P-384, and P-512.<br><br>The TOE’s implementation of RSA for digital signature generation and verification meets FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using RSASSA-PKCS1v1_5.<br><br>The TOE’s implementation of ECDSA for digital signature generation and verification meets FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D.<br><br>This is enforced by the OpenSSL cryptographic module and is not configurable by the security administrator. |
| FCS_COP.1/Hash | The TOE performs SHA-256, SHA-384, and SHA-512 hashing for utilization in HMACs, Digital Signatures, Signature Generation/Verification, and password obfuscation. The TOE implements hashing for use in RSA and ECDSA digital signatures, HMACs in SSH Server TSF, NTP Server and HMACs and Key Derivation Functions in TLS Client TSF. The TOE generates hashes with 256, 384, or 512-bit message digest size. |
| FCS_COP.1/KeyedHash | The TOE performs keyed-hash message authentication using, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, or implicit and using cryptographic key sizes of 256, 384, 512-bits and message digest sizes of 256, 384, 512-bits, with block sizes of 512-bits (for SHA-256) and 1024-bits (for SSH-384 and SHA-512), per RFC-4868. |
| FCS_RBG_EXT.1 | The TOE implements a DRBG in accordance with ISO/IEC 18031:2011 using a CTR DRBG with AES. The TSF seed the CTR_DRBG using 361-bits of data that contains at least 256 bits of entropy. The TSF gathers and pools entropy from 1 platform based noise source which is thermal noise. The CAVP details are given in Table 18.<br><br>Additional information related to entropy functionality of the TOE can be reviewed in the Entropy Assessment Report (EAR) provided as an ancillary document. |
| FCS_NTP_EXT.1 | The TOE supports the use of NTP servers for time updates with the following NTP version: NTP v4 (RFC 5905). The TOE updates its system time with authentication using SHA256 as the message digest algorithm to verify the authenticity of the timestamp and the TOE does not update the timestamps from broadcast and/or multicast addresses. The TOE supports configuration of at least three (3) NTP time sources in the Operational Environment. |

| Requirement | TSS Description |
|---|---|
| FCS_SSH_EXT.1, FCS_SSHS_EXT.1 | The TOE implements SSHv2, compliant with RFCs 4251, 4252, 4253, 4254, 4344, 5656, 6668, 8308, and 8332. There is no SSHv1 nor telnet implementation on the TOE.<br><br>The TSF supports the following authentication methods:<br>• pasword<br>• public key<br>○ rsa-sha2-256, rsa-sha2-512<br><br>The TOE ensures and verifies that the SSH client's presented public key matches one that is stored within the TOE's SSH server's authorized keys file.<br><br>SSH connections will be dropped if the TOE receives a packet larger than 262,126 bytes. Large packets are detected by the SSH implementation and dropped internal to the SSH process.<br><br>The TSF supports the following public-key algorithms: rsa-sha2-256 and rsa-sha2-512.<br><br>The TSF supports the following encryption algorithms: aes128-ctr, aes256-ctr, aes128-cbc, and aes256-cbc, aes128-gcm@openssh.com and aes256-gcm@openssh.com to ensure confidentiality of the session.<br><br>The TSF supports the following data integrity algorithms: hmac-sha2-256, hmac-sha2-512, and implicit MAC (with aes128-gcm@openssh.com and aes256-gcm@openssh.com).<br><br>The TSF supports the following key exchange algorithms: ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521.<br><br>The TSF supports the SSH Key Derivation Function (KDF) as defined in RFC 5656, Section 4.<br><br>The TSF tracks the number of bytes encrypted with each key and the time since the last rekey. The TSF initiates a rekey if 512 MB of data is encrypted with an individual key or when 1 hour has elapsed since the last rekey. |
| FCS_TLSC_EXT.1 | The TOE utilizes OpenSSL 3.0.2 to provide a TLS client for protecting the communication channel to a remote syslog audit server. The TOE only supports TLSv1.2, rejecting all other SSL/TLS versions. The following non-configurable list of ciphersuites are supported:<br>• TLS_RSA_WITH_AES_128_CBC_SHA256<br>• TLS_RSA_WITH_AES_256_CBC_ SHA256<br>• TLS_RSA_WITH_AES_128_GCM_SHA256<br>• TLS_RSA_WITH_AES_256_GCM_SHA384<br>• TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256<br>• TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384<br>• TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256<br>• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384<br>• TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256<br>• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 |

| Requirement | TSS Description |
|---|---|
| | • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256<br>• TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384<br><br>The following signature algorithms are supported by default and are not configurable by the security administrator:<br>• rsa_pkcs1 with sha256(0x0401),<br>• rsa_pkcs1with sha384(0x0501),<br>• rsa_pkcs1 with sha512(0x0601),<br>• ecdsa_secp256r1 with sha256(0x0403),<br>• ecdsa_secp384r1 with sha384(0x0503),<br>• ecdsa_secp521r1 with sha512(0x0603)<br><br>The Supported Groups Extension is presented with the following non-configurable curves/groups in the Client Hello.<br>• secp256r1, secp384r1, secp521r1,<br><br>Reference identifiers are created by the TOE using the configuration data provided by the admin in administrative CLI . The admin configures the target destination of the remote audit server using an IP address with subnet mask of the target system or an FQDN of the target system. Wildcards are not supported for IP addresses. Wildcards are supported for FQDNs only and are only accepted when the wildcard is in the left-most label/domain of the configured FQDN.<br><br>When establishing a TLS connection to the remote audit server, the TSF uses the TLS Server certificate presented by the remote audit server to verify the server's identity.<br><br>The TSF supports the following reference identifiers:<br>• CN-ID<br>• DNS-ID<br>• IPv4 address in the SAN<br><br>The TSF establishes reference identifiers for the remote audit server as follows:<br>• When the remote audit server is specified using an IP address, the TSF verifies that the IP address exactly matches a SAN IP Address field in the server certificate using the rules specified in Section 3.1 of RFC 2818. The TSF does not support IP address in the CN field and will reject the connection attempt in the case where no SAN is present and only and IP address is present in the CN.<br><br>• When the remote audit server is specified using an FQDN, the TSF verifies that the FQDN address exactly matches a DNS-ID in the server certificate using the rules specified in Section 3.1 of RFC 2818. If the server certificate does not contain the SAN, the TSF will make the comparison against the CN-ID following the rules specified in Section 3.1 of RFC 2818.<br><br>When the reference identity is an IP address, the identity is converted to the "network byte order" octet string representation. This octet string is then compared against subjectAltName value of type iPAddress (if the SAN is present). |

| Requirement | TSS Description |
|---|---|
| | A match occurs if the reference identity octet string and value octet strings are identical.<br><br>If the reference identity is an internationalized domain name, the TSF converts the value to the ASCII Compatible Encoding (ACE) format as specified in Section 4 of RFC 3490 before comparison with subjectAltName (if present) or commonName value of type dNSName. The TSF performs the conversion operation specified in Section 4 of RFC 3490 as follows:<br>• in step 1, the domain name is considered a "stored string";<br>• in step 3, the flag called "UseSTD3ASCIIRules" is set;<br>• in step 4, each label is processed with the "ToASCII" operation; and<br>• in step 5, all label separators are changed to U+002E (full stop/period).<br><br>Canonical format (RFC 3986 for IPv4) is not enforced by the TSF.<br><br>Once the TSF has verified that the presented identifiers are valid for the remote audit server, the TSF verifies the validity of the certificate as described in FIA_X509_EXT.1/Rev TSS.<br><br>Certificate pinning is not supported. The TSF will only establish the session if the presented server certificate is valid. If the server certificate is deemed invalid, the TSF terminates the TLS handshake.<br><br>This behavior is performed by default and is not configurable by the security administrator. |
| FIA_AFL.1 | The TSF can be administered through two interfaces, the local console, and SSH.<br><br>When a user connects to the local console interface, the TSF prompts the user for a username and password. The TSF does not echo characters back to the local console while the user is entering their password. The TSF checks the username/password credentials using the Linux PAM library described below. If the username/password match an authorized administrator's credentials, the user is granted access to the command line interface (CLI).<br><br>When a user connects to the SSH interface, the TSF checks to see if the user proposed public key authentication. If the client proposed public key authentication, the TSF attempts to authenticate the user using the username and the proposed SSH public key protocol (RSA-SHA2/ECDSA-SHA2). If the public key authentication fails or the client did not propose public key authentication, the TSF attempts to authenticate the client using a username/password. If either the RSA-SHA2/ECDSA-SHA2 authentication or username/password match an authorized administrator's credentials, the user is granted access to the command line interface.<br><br>For both authentication modes, the underlying Linux PAM library is used to authenticate the user in the operational environment. The TOE checks the local database for a match. For SSH public keys, the user must first authenticate using their username and password and then install their SSH public key onto the TOE. The next the time the user attempts to login, the Linux PAM library will check locally, if their public key matches. |

| Requirement | TSS Description |
|---|---|
| | The TSF supports passwords that include any character that can be entered from a standard US keyboard: upper and lower case letters, numbers, special characters [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”], and ASCII hexadecimal codes 0x20-0x2F, 0x3A-0x40, 0x5B-0x60, 0x7B-0x7E.<br><br>For each login method, a customizable banner is displayed immediately before the username/password prompt or username/public key exchange.<br><br>The password based authentication mechanism is managed by the underlying Linux operating system and the PAM library. The underlying service responsible for hashing and storing the password on the SSD will also do a pre-check (before hashing) to ensure the length of the password meets the configured minimum.<br><br>If a user enters an incorrect password enough times to meet the configured threshold for this TSF, the offending user account will be prevented from successfully authenticating until an Administrator defined time period has elapsed (a configurable range of between 1 second and 90 days). The offending account will be locked out of both the local and remote management interfaces. To ensure that administrative access is never completely locked out due to the FIA_AFL.1 functionality, the ‘pureuser’ account will, at all times, be accessible at the local console, regardless of the FIA_AFL.1 status of this account.<br><br>Failed authentication attempts are tracked by the underlying OS module PAM (Pluggable Authentication Module) using a monotonically incrementing counter. This counter is reset upon successful authentication of the offending account or after the administrator-defined time period for account lockout has elapsed. Each valid account attempting to authenticate remotely gets its own counter. This TSF only applies to remote authentication attempts. To unlock a locked account, any administrator account that is not locked can issue a command at the local or remote SSH interface to unlock a locked account.<br><br>To reiterate, the ‘pureuser’ account will always be accessible at the local console. This account can be used to unlock any locked account. |
| FIA_PMG_EXT.1 | Passwords created for user authentication to the TOE’s local and remote administrative interfaces may be composed of the following:<br>• any combination of upper and lower case letters, numbers, and the following special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, ASCII hexadecimal codes 0x20-0x2F, 0x3A-0x40, 0x5B-0x60, 0x7B-0x7E<br><br>Passwords sizes supported by the TOE range from 1 character (minimum) to 100 characters (maximum).<br><br>The minimum password length is configurable from 1 to 100 characters. Such password policies are managed by the authenticated administrator and are enforced by the Linux PAM module. The guidance documentation requires that administrators configure the minimum password length to 15 or more characters in the evaluated configuration. |

| Requirement | TSS Description |
|---|---|
| FIA_UIA_EXT.1 | The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:<br>• View the warning and consent banner in accordance with FTA_TAB.1<br>• Respond to ICMP Echo Request<br>• Respond to ARP requests with ARP replies<br>• Make DNS Requests<br><br>The TSF requires that each administrative user be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.<br><br>FIA_AFL.1 TSS of this document describes the logon process for each logon method (local, remote SSH) supported for the product. A successfully authenticated administrative user will be presented with a management interface (SSH CLI). |
| FIA_UAU.7 | At the local console, the TOE does not echo back the characters typed in for the password credential. |
| FIA_X509_EXT.1/Rev | The TOE validates x509v3 certificates according to the validation rules described in RFC 5280. Certificates presented for authentication are checked for revocation status via the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, using HTTP requests to the OCSP responder. The TOE queries an OCSP responder via the schema defined in the Authority Information Access extension (specifically the accessMethod and accessLocation fields) that is provided in the certificate to be validated – only HTTP method is supported, and only URL locations are supported. The OpenSSL module performs this certification validation functionality.<br><br>The TOE is the client in a TLS connection, the TOE will validate the peer certificate of the remote audit server during the TLS handshake, as soon as the certificate is received by the TOE. The handshake will fail if the certificate is deemed invalid by the TOE.<br><br>When a certificate is used (to identify the TSF or identify an external entity to the TOE), the TOE verifies certificates by verifying the following:<br>1. The current date is between the “Valid from” and “Valid to” dates listed in the certificate<br>2. The certificate is not revoked when a response from an OCSP responder is successfully provided to the TSF<br>3. The certificate chain is valid:<br>a. Each certificate in the certificate chain passes the checks described in #1 and #2 above.<br>b. Each certificate (other than the first certificate) in the certificate chain has the basicConstraints extension ‘Subject Type=CA’.<br>c. Each certificate is signed by:<br>i. a certificate (which has the ‘certificate signing’ key-usage extension) in the certificate chain installed on the TOE, or<br>ii. a trusted root CA that has been installed on the TOE |

| Requirement | TSS Description |
|---|---|
| | The TOE supports the following extendedKeyUsage fields (listed using Object Identifiers as found in RFC 7299 'Object Identifier Registry for the PKIX Working Group,' Section 'SMI Security for PKIX Extended Key Purpose' Registry):<br>• id-kp-serverAuth OBJECT IDENTIFIER ::= { id-kp 1 }<br>• id-kp-OCSPSigning OBJECT IDENTIFIER ::= { id-kp 9 }<br><br>The TSF only recognizes the extended key usage purposes listed above. For the extended key usage purposes that the TOE recognizes, the TOE will reject a certificate if the certificate is attempted to be used for a purpose that does not match the extended key usage purpose listed in said certificate. The TOE will reject TLS connections with certificates for all other extended key usage purposes.<br><br>While the TOE recognizes the id-kp-clientAuth purpose, there is no TSF for which this purpose would be used.<br><br>The TOE supports a minimum certificate path length of (3) three, thus at minimum a Root CA, an intermediate CAs, and an end-entity/leaf certificate are supported. |
| FIA_X509_EXT.2 | The administrative CLI supports RSA and ECDSA certificates for authentication of the TSF to a peer. The administrator can generate CSRs using the RSA or ECDSA algorithm. Administrators also have the option to import an RSA or ECDSA certificate and associated private key into the TOE. Only one Server certificate may be installed on the system at any one time. The TOE, at minimum, supports installation of a trust certificate chain of one root CA, and an intermediate CA's.<br><br>The configured/installed RSA certificate is sent to the peer when an RSA-authentication based ciphersuite is chosen by the TOE during a TLS handshake with a TLS client<br><br>When the TOE is validating a peer certificate, such as when establishing the TLS session to the remote audit server, the TOE chooses the installed/trusted CA certificate that is associated to the incoming peer certificate. If the peer certificate is signed by an unknown CA, the TOE has no certificates to choose from, failing the validation attempt and, thus, the TLS handshake will be terminated.<br><br>When an OCSP responder does not provide a response, or the OCSP responder is not available, the TOE will not accept the certificate. |
| FIA_X509_EXT.3 | Administrators are able to generate a CSR through the local and remote CLI interfaces. In addition to the RSA or ECDSA public-key data that is automatically included in the CSR, the admin can specify the following additional information in the CSR:<br>• Common Name<br>• Organization<br>• Organizational Unit<br>• Country |

| Requirement | TSS Description |
|---|---|
| FMT_MOF.1/ManualUpdate | The TOE restricts the ability to initiate manual updates of the TOE to identified and authenticated Security Administrators by virtue of restricted access to TOE's administrative interfaces. This functionality is only available via the TOE's administrative interfaces.<br><br>The administrative interfaces are only accessible to administrators after successful authentication; consequently, this functionality is not available to non-administrators. |
| FMT_MTD.1/CoreData | The TOE restricts the ability to manage the TSF data to identified and authenticated Security Administrators by virtue of restricted access to TOE's administrative interfaces. This functionality is only available via the TOE's administrative interfaces. The administrative interfaces are only accessible to administrators after successful authentication; consequently, this functionality is not available to non-administrators.<br><br>There are no interfaces to the TOE that provide the ability for an unauthenticated user to view/modify/edit TSF data. The local console and remote management interfaces only provide the warning and consent banner prior to administrative authentication to the TOE. Non-security administrative users do not have interfaces to view/modify/edit the TSF data.<br><br>As described for FMT_SMR.2 below, the TOE supports four administrative roles. Of these roles, only "Array Administrators" can create, modify, or delete TOE configuration settings (parameters). This includes generating and exporting X.509 CSRs, importing X.509 keys and or certificates, and deleting X.509 keys and certificates . The TOE does not support modifying configured X.509 certificates. |
| FMT_MTD.1/Cryptokeys | The Security Administrator is authorized to manage:<br>• X509 certificates and Certificate Authorities (CAs)<br>  ○ Generate, Import, Export, Delete<br>• SSH public keys<br>  ○ Import, Delete<br>• Passwords<br>  ○ Create, Reset<br>• NTP Symmetric Keys (SHA256 message digest algorithm)<br>  ○ Import, Delete<br><br>These keys are managed via TOE's administrative interfaces, which provides granular control over key management (ability to import SSH keys, export cryptographic keys, and delete keys). Importantly, Cryptographic keys can only be managed by identified and authenticated security administrators to the TOE.<br><br>The administrative interfaces are only accessible to administrators after successful authentication; consequently, this functionality is not available to non-administrators. They can also set up Network Time Protocol (NTP) connections utilizing a SHA256 message digest algorithm, ensuring synchronized timekeeping across devices. |
| FMT_SMF.1 | The TOE supports two administrative interfaces. One is the local administration interface which corresponds to the TOE local console. The other is the remote |

| Requirement | TSS Description |
|---|---|
| | administration interface which corresponds to access to the TOE via SSH trusted path.<br><br>Identified and authenticated security administrators have the ability to do the following:<br>• Ability to administer the TOE remotely;<br>• Ability to configure the access banner<br>  ○ When a global banner is configured it is presented at the local and remote administrative interfaces (local console and SSH respectively).<br>  ○ This is configured in either administrative interface.<br>• Ability to configure the remote session inactivity time before session termination;<br>  ○ Session inactivity time for the local and remote CLI (local console and SSH), is able to be configured via any of those interfaces.<br>• Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;<br>  ○ The update package is verified using digital signature verification.<br>  ○ This is configured on the SSH remote interface.<br>• Ability to modify the behaviour of the transmission of audit data to an external IT entity;<br>  ○ This is configured in either administrative interface.<br>• Ability to manage the cryptographic keys;<br>  ○ This is configured in either administrative interface.<br>• Ability to configure the cryptographic functionality;<br>  ○ This is configured in either administrative interface.<br>• Ability to re-enable an Administrator account;<br>  ○ This is configured in either administrative interface.<br>  ○ Note: while the TSF for FIA_AFL.1 is a time-based lockout mechanism, administrators have the ability to unlock offending accounts prior to the lockout time elapsing. Unlocking can be achieved through all administrative interfaces.<br>• Ability to configure NTP;<br>  ○ This is configured in either administrative interface.<br>• Ability to configure the reference identifier for the peer;<br>  ○ This is configured in either administrative interface.<br>• Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;<br>  ○ This is configured in either administrative interface.<br>• Ability to generate Certificate Signing Request (CSR) and process CA certificate response;<br>  ○ This is configured in either administrative interface.<br>• Ability to administer the TOE locally;<br>• Ability to configure the local session inactivity time before session termination or locking;<br>  ○ This is configured in either administrative interface.<br>• Ability to configure the authentication failure parameters for FIA_AFL.1;<br>  ○ This is configured in either administrative user interface.<br>• Ability to manage the trusted public keys database; |

| Requirement | TSS Description |
|---|---|
| | ○ This is configured in either administrative interface. |
| FMT_SMR.2 | The TOE maintains the role of 'Security Administrator.' The TOE maintains additional user roles that can be assigned to users of the TOE. The TSF includes four administrative roles within the Authorized Administrator role:<br>• Internal Administrator<br>• Array Administrator<br>• Storage Administrator<br>• Read-Only Administrator<br><br>All roles listed above are considered authorized 'Security Administrators.' All administrators may administer the TOE locally and remotely. |
| FPT_APW_EXT.1 | Locally stored passwords are stored as SHA-512 hashes (including a salt). The administrative interfaces do not provide methods to view password hashes. |
| FPT_SKP_EXT.1 | The CLI (both local and remote) uses a proprietary prompt that allows users access to pre-defined binaries created by Pure Storage. Users do not have file system access. |
| FPT_STM_EXT.1 | The TOE implements a clock to be configured to update the time from an external NTP server. The time is maintained by the TOE connecting to three (3) or more external NTP servers. If an NTP server is configured, the TOE synchronizes the time with the external NTP server periodically.<br><br>The following TSF security functions utilize the time:<br>• Audit Record timestamps<br>• SSH Console session timeout<br>• Local Console session timeout<br>• x509 Certificate expiration checking<br>• Input for the 'Random' field in the TLS Client_Hello Handshake message, as a 32-bit unsigned integer |
| FPT_TST_EXT.1 | Upon power-up or reboot, the TSF performs a SHA-1 of the kernel, all executables, and all interpreted files, ensuring that the integrity of the operating systems and executables is maintained. The TSF also performs a known answer test on each cryptographic algorithm. If all the hash integrity check passes and the cryptographic algorithms are operating correctly, the TSF will begin normal operation. These tests demonstrate the correct operation of the device by ensuring that no modifications to the operating system and executables have been made, that only tested code is being run by the TSF, and that the underlying hardware is able to load the OS and handle each known answer test correctly.<br><br>Cryptographic algorithm self-tests include:<br>• RSA -Sign, Verify<br>• ECDSA - Sign, Verify<br>• DRBG - CTR_DRBG: AES<br>• SHS - SHA-1, SHA2-512<br>• HMAC - HMAC-SHA-256<br>• AES - Encrypt, Decrypt<br>• KAS-ECC-SSC - Shared Secret (Z) Computation |

| Requirement | TSS Description |
|---|---|
| | In the event of a self-test failure, the TOE will halt start-up and display a error message on the local console until the TOE is subsequently power-cycled. |
| FPT_TUD_EXT.1 | The currently installed version of software can be queried on the TOE by running the 'purearray list' command from the CLI. Updates are initiated and installed via the internal administrator role. The administrator logs into the TOE as root, copies the update to the TOE, then proceeds to install the update using procedures provided in the guidance documentation.<br><br>The update package is signed using ECDSA with SHA-384 digital signature which is verified during the upgrade. If the verification is successful, the upgrade will be loaded onto the TOE. If the verification fails, the upgrade operation will fail, an error message will be displayed to the administrator, and the upgrade operation will be aborted.<br><br>The TSF provides Security Administrators the ability to manually initiate updates to TOE firmware/software via the local console and remotely using SSH. There are no other update mechanisms.<br><br>There are no delays to the updating of the TOE software. Once TOE update is initiated, the TOE performs the installation of the software. It is necessary to reboot in order to boot the TOE using the updated TOE fireware/software. At no point during operation is it unclear as to which version of the TOE software is installed since there is no queuing or staging of software updates.<br><br>The file names of the candidate update package follow the pattern of<br>• purity_<version>_<build>.ppkg. |
| FTA_SSL_EXT.1,<br>FTA_SSL.3,<br>FTA_SSL.4 | The TSF allows two methods of administrator access: local serial access and remote SSH. The administrator can terminate an administrative session by logging out using either the "exit" or "logout" CLI command.<br><br>Inactivity timeouts for SSH and local console CLI sessions are configured by the "purearray setattr --idle-timeout" CLI command for an administrator specified number of minutes. After the configured period of inactivity on a given remote SSH or local console CLI session, the TOE will terminate the session. |
| FTA_TAB.1 | The TSF allows two methods of administrator access: local serial access and remote SSH. The TSF displays a configurable advisory and consent message when an administrator accesses any administrative interface. The advisory message is configured and managed through an option in the administrative interface and enforced through interface-specific TSF code. |
| FTP_ITC.1 | The TSF uses OpenSSL and syslog-ng to communicate with remote audit log servers via TLS.<br><br>Each protocol implementation is performed according to the descriptions and requirements provided in FCS_CKM.1 TSS and FCS_TLSC_EXT.1 TSS above.<br><br>The TOE initiates the following Trusted Channel communications:<br>• TLS (as client) channel to remote audit server |

| Requirement | TSS Description |
|---|---|
| | ○ The TOE is assured the identity of the non-TSF endpoint via validation of the x509 peer certificate (audit server's server certificate) received in the TLS handshake to the remote audit server. Certificate validation is described in FIA_X509_EXT.1/Rev TSS above. |
| FTP_TRP.1/Admin | The TSF uses OpenSSH to provide a remote CLI interface for administrators, which is protected via SSH. Relevant ciphersuites and algorithms are enumerated in FCS_CKM.1 TSS and FCS_SSHS_EXT.1 TSS above. |

## 6.1 CAVP Algorithm Certificate Details

Each of these cryptographic algorithms have been validated as identified in the table below.

**Table 18 – CAVP Algorithm Certificate References**

| Algorithm | CAVP Cert # | Standard | Operation | SFR |
|---|---|---|---|---|
| RSA | A6623 | FIPS 186-4 | Key Generation<br>Signature Generation/ Verification<br><br>Mod lengths: 2048 (bits) | FCS_CKM.1<br>FCS_CKM.2.1<br>FCS_COP.1/SigGen |
| ECDSA | A6623 | FIPS 186-4<br>SP 800-56A Revision 3 | Key Generation / Validation<br>Signature Generation/ Verification<br><br>Curves: P-256, P-384, P-521 | FCS_CKM.1<br>FCS_CKM.2.1<br>FCS_COP.1/SigGen |
| SP 800-90 DRBG | A6623 | SP 800-90A | Random Bit Generation<br><br>CTR_DRBG (AES-256) | FCS_RBG_EXT.1 |
| SHS | A6623 | ISO/IEC 10118-3:2004 | Hashing<br><br>SHA-256, SHA-384, SHA-512 | FCS_COP.1/Hash |
| HMAC-SHS | A6623 | ISO/IEC 9797-2:2011 | Keyed-Hashing<br><br>HMAC-SHA-256,<br>HMAC-SHA-384,<br>HMAC-SHA-512 | FCS_COP.1/ KeyedHash |
| AES | A6623 | AES specified in ISO 18033-3<br>CBC specified in ISO 10116<br>CTR as specified in ISO 10116<br>GCM specified in ISO 19772 | Encryption/ Decryption<br><br>CBC, CTR, GCM<br>Key Lengths: 128, 256 | FCS_COP.1/ DataEncryption |
| KAS-ECC-SSC | A6623 | SP 800-56A Revision 3 | Key Establishment<br><br>Curves: P-256, P-384, P-521 | FCS_CKM.2.1 |

## 6.2 Cryptographic Key Destruction

The table below describes the key zeroization provided by the TOE and as referenced in FCS_CKM.4.

**Table 19 – Keys/CSPs Destruction**

| Keys/CSPs | Purpose | Storage Location | Method of Zeroization |
|---|---|---|---|
| EC Diffie Hellman private parameters | ECDH Key | RAM | One-pass overwrite with zeroes. |
| EC Diffie Hellman public parameters | ECDH Key | RAM | One-pass overwrite with zeroes. |
| ECDH Shared Secret | ECDH Key | RAM | One-pass overwrite with zeroes. |
| Diffie Hellman private parameters | DH Key | RAM | One-pass overwrite with zeroes. |
| Diffie Hellman public parameters | DH Key | RAM | One-pass overwrite with zeroes. |
| DH Shared Secret | DH Key | RAM | One-pass overwrite with zeroes. |
| SSH Private Keys | RSA Private Keys | Local filesystem | Three-pass overwrite with random pattern |
| SSH Public Keys | RSA Public Keys | n/a - public | Three-pass overwrite with random pattern |
| SSH Private Keys | ECDSA Private Keys | Local filesystem | Three-pass overwrite with random pattern |
| SSH Public Keys | ECDSA Public Keys | n/a - public | Three-pass overwrite with random pattern |
| TLS Private Keys | RSA Private Keys | Local filesystem | Three-pass overwrite with random pattern |
| TLS Public Keys | RSA Public Keys | n/a - public | Three-pass overwrite with random pattern |
| TLS Private Keys | ECDSA Private Keys | Local filesystem | Three-pass overwrite with random pattern |
| TLS Public Keys | ECDSA Public Keys | n/a - public | Three-pass overwrite with random pattern |
| SSH Session Encryption Keys | AES Keys | RAM | One-pass overwrite with zeroes. |
| SSH Session Integrity Keys | HMAC Keys | RAM | One-pass overwrite with zeroes. |
| TLS Session Encryption Keys | AES Keys | RAM | One-pass overwrite with zeroes. |
| TLS Session Integrity Keys | HMAC Keys | RAM | One-pass overwrite with zeroes. |

# 7 Acronym Table

Acronyms should be included as an Appendix in each document.

**Table 20 – Acronyms**

| Acronym | Definition |
|---|---|
| **AES** | Advanced Encryption Standard |
| **CC** | Common Criteria |
| **CRL** | Certificate Revocation List |
| **ECDSA** | Elliptic Curve Digital Signature Algorithm |
| **EP** | Extended Package |
| **IP** | Internet Protocol |
| **NDcPP** | Network Device Collaborative Protection Profile |
| **NIAP** | Nation Information Assurance Partnership |
| **NTP** | Network Time Protocol |
| **OCSP** | Online Certificate Status Protocol |
| **PP** | Protection Profile |
| **RSA** | Rivest, Shamir, & Adleman |
| **SFR** | Security Functional Requirement |
| **SSH** | Secure Shell |
| **ST** | Security Target |
| **TOE** | Target of Evaluation |
| **TLS** | Transport Layer Security |
| **TSS** | TOE Summary Specification |