FlashArray//X running Purity//FA 5.3 Security Target

18-4177-R-0012

Version: 3.2

January 7, 2021

Prepared For:

Pure Storage, Inc.

650 Castro Street, Suite #260

Mountain View, CA 94041

Prepared By:

Michael C. Baron

UL Verification Services Inc.

Notices:

Table of Contents

1. Security Target (ST) Introduction

The structure of this document is defined by CC v3.1r5 Part 1 Annex A.2, “Mandatory contents of an ST”:

- Section 1 contains the ST Introduction, including the ST reference, Target of Evaluation (TOE) reference, TOE overview, and TOE description.
- Section 2 contains conformance claims to the Common Criteria (CC) version, Protection Profile (PP) and package claims, as well as rationale for these conformance claims.
- Section 3 contains the security problem definition, which includes threats, Organizational Security Policies (OSP), and assumptions that must be countered, enforced, and upheld by the TOE and its operational environment.
- Section 4 contains statements of security objectives for the TOE, and the TOE operational environment as well as rationale for these security objectives.
- Section 5 contains definitions of any extended security requirements claimed in the ST.
- Section 6 contains the security function requirements (SFR), the security assurance requirements (SAR), as well as the rationale for the claimed SFR and SAR.
- Section 7 contains the TOE summary specification, which includes the detailed specification of the IT security functions

1.1 Security Target Reference

The Security Target reference shall uniquely identify the Security Target.

ST Title:	FlashArray//X running Purity//FA 5.3 Security Target
ST Version Number:	3.2
ST Author(s):	Michael C. Baron
ST Publication Date:	January 7, 2021
Keywords	Network Device

1.2 Target of Evaluation Reference

The Target of Evaluation reference shall identify the Target of Evaluation.

TOE Developer	Pure Storage, Inc. 650 Castro Street, Suite #260 Mountain View, CA 94041
TOE Name	FlashArray//X running Purity//FA 5.3
TOE Hardware identifiers	FlashArray//X R2 Family: • X10 R2 • X20 R2 • X50 R2 • X70 R2

	• X90 R2 FlashArray//X R3 Family: • X10 R3 • X20 R3 • X50 R3 • X70 R3 • X90 R3
TOE Software identifier	Purity//FA 5.3.2.post10

1.3 Target of Evaluation Overview

1.3.1 TOE Product Type

The TOE is classified as a Network Device.

1.3.2 TOE Usage

Pure Storage Inc's (Pure Storage) FlashArray//X R2 and R3 (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//X R2 and R3 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 HTTPS/TLS with cryptographic encryption and authentication using FIPS 140-2 approved algorithms. The TOE also supports use of external audit servers, protected by TLS.

NTP functionality is unevaluated; Security Administrative users are instructed to disable NTP functionality in the evaluated configuration.

The TOE is not a distributed TOE.

1.3.3 TOE Major Security Features Summary

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

1.3.4 TOE IT environment hardware/software/firmware requirements

1.3.4.1 Network/Software Requirements

The following elements must be present in the Operational Environment of the TOE to meet the SFRs:

Syslog Server:

- Conforming to the following RFCs:
  - RFC 3164 - The BSD syslog Protocol
  - RFC 5425 - TLS Transport Mapping
- Supporting the following for TLS:
  - Protocol versions:
    - TLSv1.2 (Conforming to RFC 5246)
  - Supporting at least one of the following TLS Ciphersuites:
    - TLS_RSA_WITH_AES_128_CBC_SHA
    - TLS_RSA_WITH_AES_256_CBC_SHA
    - TLS_RSA_WITH_AES_128_CBC_SHA256
    - TLS_RSA_WITH_AES_256_CBC_SHA256
    - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
    - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
    - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
    - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
    - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
    - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
    - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

Remote Administrative User Access:

- For remote administrative user access, the TOE requires the following to be present in the OE:
  - Utilization of a web browser supporting the following details:
    - Protocol versions (at least one of):
      - HTTPS/TLSv1.2 (Conforming to RFCs 2818 & 5246)
    - Supporting at least one of the following TLS Ciphersuites:
      - TLS_RSA_WITH_AES_128_CBC_SHA
      - TLS_RSA_WITH_AES_256_CBC_SHA
      - TLS_DHE_RSA_WITH_AES_128_CBC_SHA
      - TLS_DHE_RSA_WITH_AES_256_CBC_SHA
      - TLS_RSA_WITH_AES_128_CBC_SHA256
      - TLS_RSA_WITH_AES_256_CBC_SHA256
      - TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
      - TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
      - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
      - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
      - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
      - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
      - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
      - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
      - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
      - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

- The TOE is known to be compatible with the following web browsers:
  - Chrome version 67 and newer
  - Apple Safari version 11 and newer
  - Mozilla Firefox 61 and newer
  - Microsoft Edge versions 42 and newer
  - Microsoft Internet Explorer version 11 and newer

- An SSH client supporting the following:
  - The TOE requires an SSH client supporting:
    - Protocol versions:
      - SSHv2
        - Conforming to RFCs 4251-4254, 5656, and 6668
    - Public-Key Algorithms:
      - ssh-rsa
      - rsa-sha2-256
      - rsa-sha2-512
    - Data Encryption (at least one of the following):
      - AES-CBC-128
      - AES-CBC-256
      - AES128-CTR
      - AES256-CTR
      - aes128-gcm@openssh.com
      - aes256-gcm@openssh.com
    - Data Integrity (at least one of the following):
      - hmac-sha1
      - hmac-sha2-256
      - hmac-sha2-512
      - “Implicit”
    - Key Exchange (at least one of the following):
      - diffie-hellman-group14-sha1
      - ecdh-sha2-nistp256
      - ecdh-sha2-nistp384
      - ecdh-sha2-nistp521

1.3.4.2 Hardware Requirements

Local Console:

- VGA Monitor
- USB Mouse and Keyboard (HID-compliant)

SAS-connected SSD Storage Array from Pure Storage

1.4 Target of Evaluation Description

1.4.1 Target of Evaluation Physical Boundaries

Hardware:

The TOE consists of the following FlashArray//X (R2 and R3 families) hardware models:

FlashArray//X R2 Family:

- X10 R2
- X20 R2

- X50 R2
- X70 R2
- X90 R2

FlashArray//X R3 Family:

- X10 R3
- X20 R3
- X50 R3
- X70 R3
- X90 R3

FlashArray//X R2 Family models and specifications:

	X10 R2	X20 R2	X50 R2	X70 R2	X90 R2
CPU	Intel® Xeon® Silver 4108 Processor	Intel® Xeon® Silver 4114 Processor	Intel® Xeon® Silver 4116 Processor	Intel® Xeon® Gold 6130 Processor	Intel® Xeon® Gold 6152 Processor
Microarchitecture	Skylake	Skylake	Skylake	Skylake	Skylake
Total Volatile Memory	96GB	192GB	192GB	384GB	768GB
Management Ports	2 x 1Gb Management Ports 1x Local Console Port	2 x 1Gb Management Ports 1x Local Console Port	2 x 1Gb Management Ports 1x Local Console Port	2 x 1Gb Management Ports 1x Local Console Port	2 x 1Gb Management Ports 1x Local Console Port

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® Gold 6252 Processor
Microarchitecture	Cascade Lake	Cascade Lake	Cascade Lake	Cascade Lake	Cascade Lake
Total Volatile Memory	96GB	192GB	288GB	384GB	768GB
Management Ports	2 x 1Gb Management Ports 1x Local Console Port	2 x 1Gb Management Ports 1x Local Console Port	2 x 1Gb Management Ports 1x Local Console Port	2 x 1Gb Management Ports 1x Local Console Port	2 x 1Gb Management Ports 1x Local Console Port

TOE Software:

Each model of the TOE runs the following Operating System software:

- Purity//FA 5.3.2.post10

Guidance Documentation:

The guidance documentation that is part of the TOE is listed in Section 9 “References” within Table 10: TOE Guidance Documentation.

Format and Delivery of the TOE:

The physical portion of the TOE (hardware) is delivered to the consumer via currier services. The hardware is delivered with the evaluated version of the TOE software pre-installed.

For access to evaluated version of the software (for backup or reinstallation purposes, etc.), and for access to future Trusted Update software packages, Pure Storage provides CC customers with authentication credentials to log in to the Pure Storage support web portal. From this portal, the customer navigates to the ‘Downloads’ section, and downloads the necessary software installation package. This software installation package is a binary file.

The guidance documentation is of PDF format and is available to the customer by download from the Pure Storage support web portal.

1.4.2 Target of Evaluation Description

The logical boundary of the TOE includes those security functions implemented exclusively by the TOE. These security functions are listed in Section 1.3.3 above and are further described in the following subsections. A more detailed description of the implementation of these security functions is provided in Section 7 “TOE Summary Specification”.

1.4.2.1 Audit

- The TOE will audit all events and information defined in Table 3: 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.
- The TOE can transmit audit data to an external IT entity using the Syslog over TLS protocol.

1.4.2.2 Cryptographic Operations

The TSF performs the following cryptographic operations:

- SSH for remote CLI administrative management of the TOE:
    - Protocol versions:
        - SSHv2
            - Conforming to RFCs 4251-4254, 5656, and 6668
    - Public-Key Algorithms:
        - ssh-rsa
            - 2048-bit RSA keys
    - Data Encryption:
        - AES-CBC-128
            - 128-bit AES symmetric key
        - AES-CBC-256
            - 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 Integrity:
      - hmac-sha1
      - hmac-sha2-256
      - hmac-sha2-512
      - “Implicit”
    - Key Exchange:
      - diffie-hellman-group14-sha1
      - ecdh-sha2-nistp256
      - ecdh-sha2-nistp384
      - ecdh-sha2-nistp521
- HTTPS for remote administrative management of the TOE:
    - Protocol versions supporting:
      - HTTPS/TLSv1.2 (Conforming to RFCs 2818 & 5246)
    - Supporting the following TLS Ciphersuites:
      - TLS_RSA_WITH_AES_128_CBC_SHA
      - TLS_RSA_WITH_AES_256_CBC_SHA
      - TLS_DHE_RSA_WITH_AES_128_CBC_SHA
      - TLS_DHE_RSA_WITH_AES_256_CBC_SHA
      - TLS_RSA_WITH_AES_128_CBC_SHA256
      - TLS_RSA_WITH_AES_256_CBC_SHA256
      - TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
      - TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
      - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
      - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
      - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
      - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
      - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
      - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
      - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
      - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

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

1.4.2.3 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
  - Respond to HTTP Get Requests on TCP port 80 with a HTTP 301 'Moved Permanently' Status Code redirecting to TCP port 443
  - Respond to TLS Client_Hello messages with TLS Server_Hello messages on TCP port 443

1.4.2.4 Security Management

- TSF data includes the following:
  - All audit records generated to meet the auditing requirements of the PP
  - 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 HTTPS (remote), SSH (remote), and a local console.
- The TOE authenticates administrative users using a username/password combination or a username/SSH_RSA 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 published hash verification.

1.4.2.5 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.
- The TOE provides a means to verify firmware/software updates to the TOE using a published hash mechanism to verify the candidate update package prior to installing the update.

1.4.2.6 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, and HTTPS remote web GUI 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.

1.4.2.7 Trusted Path/Channels

- The TOE uses TLS to provide a trusted communication channel between itself and all authorized IT entities that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.
- 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 HTTPS protected trusted path, as well as 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.

1.5 Notation, formatting, and conventions

The notation, formatting, and conventions used in this Security Target are defined below; these styles and clarifying information conventions were developed to aid the reader.

Where necessary, the ST author has added application notes to provide the reader with additional details to aid understanding; they are italicized and usually appear following the element needing clarification.

The notation conventions that refer to iterations, assignments, selections, and refinements made in this Security Target are in reference to SARs and SFRs taken directly from CC Part 2 and Part 3 as well as any SFRs and SARs taken from a Protection Profile.

The notation used in those PP to indicate assignments, selections, and refinements of SARs and SFRs taken from CC Part 2 and Part 3 is not carried forward into this document. Additionally, obvious errors in the PP are corrected and noted as such.

The CC permits four component operations (assignment, iteration, refinement, and selection) to be performed on requirement components. These operations are defined in Common Criteria, Part 1; Section 8.1, "Operations" as:

- Iteration: allows a component to be used more than once with varying operations;
- Assignment: allows the specification of parameters;
- Selection: allows the specification of one or more items from a list; and
- Refinement: allows the addition of details.

Iterations performed by the ST author are indicated by alphanumeric characters within in parenthesis following the requirement number, e.g., FIA_UAU.1.1(abc123); the iterated requirement titles are similarly indicated, e.g., FIA_UAU.1(1). Iterations performed by the PP author are indicated by a slash followed by a short description, e.g. FCS_COP.1/Hash.

Assignments are identified with **bold text.**

Selections are identified with <u>underlined text</u>**.**

Refinements that add text use ***bold and italicized text*** to identify the added text***.*** Refinements that performs a deletion, identifies the deleted text with ~~***strikeout, bold, and italicized text***~~.

2. Conformance Claims

2.1 Common Criteria Conformance Claims

The TOE and Security Target are conformant to the Common Criteria Version 3.1r5, CC Part 2 extended [C2], and CC Part 3 conformant [C3].

2.2 Conformance to Protection Profiles

The TOE and Security Target claim exact conformance to the collaborative Protection Profile for Network Devices, Version 2.1, dated September 24, 2018 [cPP]. This Protection Profile will be referred to as cPP or PP for convenience throughout this Security Target.

The TOE conforms with the following Technical Decisions:

Table 1: Applied Technical Decisions

TD	TD Title
0536	NIT Technical Decision for Update Verification Inconsistency
0532	NIT Technical Decision for Use of seeds with higher entropy
0531	NIT Technical Decision for Challenge-Response for Authentication
0530	NIT Technical Decision for FCS_TLSC_EXT.1.1 5e test clarification
0529	NIT Technical Decision for OCSP and Authority Information Access extension
0484	NIT Technical Decision for Interactive sessions in FTA_SSL_EXT.1 & FTA_SSL.3
0483	NIT Technical Decision for Applicability of FPT_APW_EXT.1
0482	NIT Technical Decision for Identification of usage of cryptographic schemes
0481	NIT Technical Decision for FCS_(D)TLSC_EXT.X.2 IP addresses in reference identifiers
0480	NIT Technical Decision for Granularity of audit events
0478	NIT Technical Decision for Application Notes for FIA_X509_EXT.1 iterations
0477	NIT Technical Decision for Clarifying FPT_TUD_EXT.1 Trusted Update
0475	NIT Technical Decision for Separate traffic consideration for SSH rekey
0450	NIT Technical Decision for RSA-based ciphers and the Server Key Exchange message
0425	NIT Technical Decision for Cut-and-paste Error for Guidance AA
0424	NIT Technical Decision for NDcPP v2.1 Clarification - FCS_SSHC/S_EXT1.5
0423	NIT Technical Decision for Clarification about application of Rfl#201726rev2
0412	NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy
0410	NIT technical decision for Redundant assurance activities associated with FAU_GEN.1
0409	NIT decision for Applicability of FIA_AFL.1 to key-based SSH authentication
0408	NIT Technical Decision for local vs. remote administrator accounts
0402	NIT Technical Decision for RSA-based FCS_CKM.2 Selection
0401	NIT Technical Decision for Reliance on external servers to meet SFRs
0400	NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment

Table 1: Applied Technical Decisions	
TD	TD Title
0399	NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2)
0398	NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR
0397	NIT Technical Decision for Fixing AES-CTR Mode Tests
0396	NIT Technical Decision for FCS_TLSC_EXT.1.1, Test 2

2.3 Conformance to Security Packages

This Security Target does not claim conformance to any security function requirements or security assurance requirements packages, neither as package-conformant or package-augmented.

2.4 Conformance Claims Rationale

To demonstrate that exact conformance is met, this rationale shows all threats are addressed, all OSP are satisfied, no additional assumptions are made, all objectives have been addressed, and all SFRs and SARs have been instantiated.

The following address the completeness of the threats, OSP, and objectives, limitations on the assumptions, and instantiation of the SFRs and SARs:

- Threats
  - All threats defined in the cPP;
  - No additional threats have been defined in this ST.
- Organizational Security Policies
  - All OSP defined in the cPP are carried forward to this ST;
  - No additional OSPs have been defined in this ST.
- Assumptions
  - All assumptions defined in the cPP for a standalone TOE are carried forward to this ST;
  - No additional assumptions for the operational environment have been defined in this ST.
- Objectives
  - All objectives defined in the cPP for a standalone TOE are carried forward to this ST. Optional and selection based SFRs defined in the cPP are carried forward to this Security Target as required by the cPP.
- All mandatory SFRs and SARs defined in the cPP are carried forward to this Security Target.

Rationale presented in the body of this ST shows all assumptions on the operational environment have been upheld, all the OSP are enforced, all defined objectives have been met and these objectives counter the defined threats.

Additionally, all SFRs and SARs defined in the cPP have been properly instantiated in this Security Target; therefore, this ST shows exact compliance to the cPP.

3. Security Problem Definition

3.1 Threats

The following section defines the security threats for the TOE, characterized by a threat agent, an asset, and an adverse action of that threat agent on that asset. These threats are taken directly from the PP unchanged.

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.

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.

T.PASSWORD_CRACKING

Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic and may allow them to take advantage of any trust relationships with other network devices.

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 Organizational Security Policies

The following section defines the organizational security policies which are a set of rules, practices, and procedures imposed by an organization to address its security needs. These threats are taken directly from the PP unchanged.

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.

3.3 Assumptions

This section describes the assumptions on the operational environment in which the TOE is intended to be used. It includes information about the physical, personnel, and connectivity aspects of the environment. The operational environment must be managed in accordance with the provided guidance documentation. The following table defines specific conditions that are assumed to exist in an environment where the TOE is deployed. These assumptions are taken directly from the PP unchanged.

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 and/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 will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will 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.

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 being 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.

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).

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.

4. Security Objectives

4.1 Security Objectives for the Operational Environment

OE.PHYSICAL

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

OE.NO_GENERAL_PURPOSE

There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

OE.NO_THRU_TRAFFIC_PROTECTION

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

OE.TRUSTED_ADMIN

Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

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

OE.UPDATES

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

OE.ADMIN_CREDENTIALS_SECURE

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

OE.RESIDUAL_INFORMATION

The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

5. Extended Components Definition

This section provides definition of the extended security functional and assurance requirements; the components that are CC Part 2 extended, and CC Part 3 extended, i.e., NIAP interpreted requirements, and extended requirements.

5.1 Extended Security Functional Requirements Definitions

There are no extended Security Functional Requirements defined in this Security Target. All extended SFRs were taken from the cPP.

5.2 Extended Security Assurance Requirement Definitions

There are no extended Security Assurance Requirements defined in this Security Target. All extended SARs were taken from the cPP.

6. Security Requirements

This section describes the security functional and assurance requirements for the TOE; those that are CC Part 2 conformant, CC Part 2 extended, CC Part 3 conformant, and CC Part 3 extended.

6.1 Security Function Requirements

This section describes the functional requirements for the TOE. The security functional requirement components in this security target are CC Part 2 conformant or CC Part 2 extended as defined in Section 2, Conformance Claims. Operations that were performed in the cPP are not signified in this section. Operations performed by the ST are denoted according to the formatting conventions in Section 1.5.

Table 2: Security Functional Requirements

SFR	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG.1	Protected audit trail storage
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation (Refinement)
FCS_CKM.2	Cryptographic Key Establishment (Refinement)
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_HTTPS_EXT.1	HTTPS Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSHS_EXT.1	SSH Server
FCS_TLSC_EXT.1	TLS Client Protocol
FCS_TLSS_EXT.1	TLS Server Protocol
FIA_AFL.1	Authentication Failure Management (Refinement)
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1/Rev	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FIA_X509_EXT.3	X.509 Certificate Requests
FMT_MOF.1/Services	Management of security functions behaviour

Table 2: Security Functional Requirements	
SFR	Description
FMT_MOF.1 /Functions	Management of security functions behaviour
FMT_MOF.1 /ManualUpdate	Management of security functions behaviour
FMT_MTD.1 /CoreData	Management of TSF Data
FMT_MTD.1 /CryptoKeys	Management of TSF data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on security roles
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
FPT_TST_EXT.1	TSF Testing (Extended)
FPT_TUD_EXT.1	Trusted Update
FPT_STM_EXT.1	Reliable Time Stamps
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination (Refinement)
FTA_SSL.4	User-initiated Termination (Refinement)
FTA_TAB.1	Default TOE Access Banners (Refinement)
FTP_ITC.1	Inter-TSF trusted channel (Refinement)
FTP_TRP.1/Admin	Trusted Path (Refinement)

6.1.1 Security Audit (FAU)

6.1.1.1 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 user account shall be logged if individual user accounts are required for Administrators).
   - Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
   - Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
   - Resetting passwords (name of related user account shall be logged).
   - no other actions;
d) Specifically defined auditable events listed in Table 3.

FAU_GEN.1.2

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

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

Table 3: Auditable Events		
SFR	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG.1	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_RBG_EXT.1	None.	None.
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store

Table 3: Auditable Events		
SFR	Auditable Events	Additional Audit Record Contents
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Services	None.	None.
FMT_MOF.1/Functions	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	All management activities of TSF data.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path.	None.

Table 3: Auditable Events		
SFR	Auditable Events	Additional Audit Record Contents
	Termination of the trusted path. Failure of the trusted path functions.	

6.1.1.2 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.

6.1.1.3 FAU_STG.1 Protected audit trail storage

FAU_STG.1.1

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

FAU_STG.1.2

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

6.1.1.4 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.

- TOE shall consist of a single standalone component that stores audit data locally,

.

FAU_STG_EXT.1.3(Method1)

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.1.3(Method2)

The TSF shall overwrite previous audit records according to the following rule: **every hour check repository capacity based on hard-coded threshold for TSS specified number of audit record entries; if threshold is met or exceeded, the TSF deletes all existing audit records in the specified repository, allowing for new audit records to be written to the specified repository** when the local storage space for audit data is full.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation (Refinement) [TD0400[1]]

FCS_CKM.1.1

The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm:

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;
- ECC schemes using "NIST curves" P-256, P-384, P-521 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;
- FFC Schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3.

6.1.2.2 FCS_CKM.2 Cryptographic Key Establishment (Refinement) [TD0402][2]

FCS_CKM.2.1

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:

- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1;
- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";
- Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3;.

6.1.2.3 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, destruction of reference to the key directly followed by a request for garbage collection;
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that
    - logically addresses the storage location of the key and performs a single overwrite consisting of zeros, a new value of the key;

that meets the following: No Standard.

[1] The application note for this SFR was modified.

[2] The first SFR selection item was modified.

6.1.2.4 FCS_COP.1/DataEncryption Cryptographic Operation (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.

6.1.2.5 FCS_COP.1/SigGen Cryptographic Operation (Signature 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 and cryptographic key sizes (modulus) **2048-bits**
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes **256, 384, 521 bits**

that meet the following:

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

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

FCS_COP.1.1/Hash

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

6.1.2.7 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-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 and cryptographic key sizes **160, 256, 384, 512-bits** and message digest sizes 160, 256, 384, 512 bits that meet the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2".

6.1.2.8 FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1.1

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

FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3

If a peer certificate is presented, the TSF shall not require client authentication if the peer certificate is deemed invalid.

6.1.2.9 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** hardware-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.

6.1.2.10 FCS_SSHS_EXT.1 SSH Server

FCS_SSHS_EXT.1.1 [TD0398[3]]

The TSF shall implement the SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, 5656, 6668, 8332.

FCS_SSHS_EXT.1.2

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

FCS_SSHS_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than **262,144** bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4

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

FCS_SSHS_EXT.1.5 [TD0424[4]]

The TSF shall ensure that the SSH public-key based authentication implementation uses ssh-rsa, rsa-sha2-256, rsa-sha2-512 as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6

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

[3] This SFR was updated by TD0398.

[4] This SFR was updated by TD0424.

FCS_SSHS_EXT.1.7

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

FCS_SSHS_EXT.1.8 [TD0475[5]]

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

6.1.2.11 FCS_TLSC_EXT.1 TLS Client Protocol (Selection-based)

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

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- 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

.

FCS_TLSC_EXT.1.2 [TD0481[6]]

The TSF shall verify that the presented identifiers of the following types: identifiers defined in RFC 6125, IPv4 address in SAN are matched to reference identifiers.

FCS_TLSC_EXT.1.3

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

- Not implement any administrator override mechanism

.

[5] This SFR was updated by TD0475.

[6] This SFR was modified by TD0481.

FCS_TLSC_EXT.1.4

The TSF shall present the Supported Elliptic Curves Extension with the following NIST curves: secp256r1, secp384r1 and no other curves in the Client Hello.

6.1.2.12 FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1

The TSF shall implement TLS 1.2 (RFC 5246) and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- 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

.

FCS_TLSS_EXT.1.2

The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and TLS 1.1.

FCS_TLSS_EXT.1.3

The TSF shall perform RSA key establishment with key size 2048 bits; generate EC Diffie-Hellman parameters over NIST curves secp256r1 and no other curves; generate Diffie-Hellman parameters of size 2048, bits.

6.1.3 Identification and Authentication (FIA)

6.1.3.1 FIA_AFL.1 Authentication Failure Management (Refinement) [TD0408][7]

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 remote Administrator from successfully authenticating until **unlock action** is taken by a local Administrator, prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed.

6.1.3.2 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 **1** and **100** characters.

6.1.3.3 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**
- **Respond to HTTP Get Requests on TCP port 80 with a HTTP 301 'Moved Permanently' Status Code redirecting to TCP port 443**
- **Respond to TLS Client Hello messages with TLS Server Hello messages on TCP port 443**

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.

[7] The SFR text and Application Note were modified.

6.1.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism [TD0408][8]

FIA_UAU_EXT.2.1

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

6.1.3.5 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.

6.1.3.6 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 certificate path validation supporting a minimum path length of three certificates.
- The certificate path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using 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 TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

FIA_X509_EXT.1.2/Rev

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

6.1.3.7 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 HTTPS, TLS and no additional uses.

FIA_X509_EXT.2.2

[8] The SFR text was modified and the Application Note was appended.

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

6.1.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1

The TSF shall generate a Certificate Request Message 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.

6.1.4 Security Management (FMT)

6.1.4.1 FMT_MOF.1/Functions Management of security functions behaviour

FMT_MOF.1.1/Functions

The TSF shall restrict the ability to determine the behaviour of, modify the behaviour of the functions transmission of audit data to an external IT entity to Security Administrators.

6.1.4.2 FMT_MOF.1/ManualUpdate Management of security functions behaviour

FMT_MOF.1.1/ManualUpdate

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

6.1.4.3 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.

6.1.4.4 FMT_MTD.1/CryptoKeys Management of TSF data (Selection-based)

FMT_MTD.1.1/CryptoKeys

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

6.1.4.5 FMT_SMF.1 Specification of Management Functions [TD0408][9]

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions:

- Ability to administer the TOE locally and remotely;

[9] The application note was modified.

- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using hash comparison capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- Ability to start and stop services;
- Ability to configure audit behaviour;
- Ability to manage the cryptographic keys;
- Ability to configure the cryptographic functionality;
- Ability to re-enable an Administrator account;
- Ability to set the time which is used for time-stamps;
- Ability to configure the reference identifier for the peer;
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
- Ability to import X.509v3 certificates to the TOE's trust store;

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

are satisfied.

6.1.4.7 FMT_MOF.1/Services Management of security functions behaviour

FMT_MOF.1.1/Services

The TSF shall restrict the ability to enable and disable start and stop services to Security Administrators.

6.1.5 Protection of the TSF (FPT)

6.1.5.1 FPT_APW_EXT.1 Protection of Administrator Passwords [TD0483[10]]

FPT_APW_EXT.1.1

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

FPT_APW_EXT.1.2

[10] This SFR was updated by TD0483.

The TSF shall prevent the reading of plaintext administrative passwords.

6.1.5.2 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.

6.1.5.3 FPT_TST_EXT.1 TSF Testing (Extended)

FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests during initial start-up (on power on) to demonstrate the correct operation of the TSF: **cryptographic self-tests, firmware integrity self-test**.

6.1.5.4 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 a means to authenticate firmware/software updates to the TOE using a published hash prior to installing those updates.

6.1.5.5 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 allow the Security Administrator to set the time.

6.1.6 TOE Access (FTA)

6.1.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1[TD0484[11]]

[11] The Application Note for this SFR was modified by TD0484.

The TSF shall, for local interactive sessions,

- terminate the session

after a Security Administrator-specified time period of inactivity.

6.1.6.2 FTA_SSL.3 TSF-initiated Termination (Refinement)

FTA_SSL.3.1 [TD0484[12]]

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

6.1.6.3 FTA_SSL.4 User-initiated Termination (Refinement)

FTA_SSL.4.1

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

6.1.6.4 FTA_TAB.1 Default TOE Access Banners (Refinement)

FTA_TAB.1.1

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

6.1.7 Trusted path/channels (FTP)

6.1.7.1 FTP_ITC.1 Inter-TSF trusted channel (Refinement)

FTP_ITC.1.1

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

FTP_ITC.1.2

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

FTP_ITC.1.3

The TSF shall initiate communication via the trusted channel for **remote audit server**.

6.1.7.2 FTP_TRP.1/Admin Trusted Path (Refinement)

FTP_TRP.1.1/Admin

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

[12] The Application Note for this SFR was modified by TD0484.

FTP_TRP.1.2/Admin

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

FTP_TRP.1.3/Admin

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

6.2 Security Assurance Requirements

This Security Target is conformant with the assurance requirements specified in the cPP.

Table 4: Assurance Requirements	
Assurance Class	Assurance Component
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life cycle support (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
Tests (ATE)	Independent testing – conformance (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

6.2.1 Extended Security Assurance Requirements

These requirements are taken directly from the cPP.

6.2.1.1 ASE: Security Target

The ST is evaluated as per ASE activities defined in the CEM. In addition, there may be Evaluation Activities specified within the SD that call for necessary descriptions to be included in the TSS that are specific to the TOE technology type.

Appendix D provides a description of the information expected to be provided regarding the quality of entropy in the random bit generator.

The TOE summary specification shall describe how the TOE meets each SFR. In the case of entropy analysis the TSS is used in conjunction with required supplementary information on Entropy.

The requirements for exact conformance of the Security Target are described in section 2 and in [SD, 3.1].

6.2.1.2 ADV: Development

The design information about the TOE is contained in the guidance documentation available to the end user as well as the TSS portion of the ST, and any required supplementary information required by this cPP that is not to be made public.

6.2.1.2.1 Basic Functional Specification (ADV_FSP.1)

The functional specification describes the TOE Security Functions Interfaces (TSFIs). It is not necessary to have a formal or complete specification of these interfaces. Additionally, because TOEs conforming to this cPP will necessarily have interfaces to the Operational Environment that are not directly invokable by TOE users, there is little point specifying that such interfaces be described in and of themselves since only indirect testing of such interfaces may be possible. For this cPP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation. No additional "functional specification" documentation is necessary to satisfy the Evaluation Activities specified in the SD.

The Evaluation Activities in the SD are associated with the applicable SFRs; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.

Evaluation Activities

The evaluator shall check the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g., audit review or performing updates). Additionally, those interfaces that are identified in the ST, or guidance documentation, as adhering to the security policies (as presented in the SFRs), are also considered security relevant. The intent, is that these interfaces will be adequately tested, and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied.

The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant.

The documents to be examined for this assurance component in an evaluation are therefore the Security Target, AGD documentation, and any supplementary information required by the cPP for aspects such as entropy analysis or cryptographic key management architecture[13]: no additional "functional specification" documentation is necessary to satisfy the Evaluation Activities. The interfaces that need to be evaluated are also identified by reference to the assurance activities listed for each SFR, and are expected to be identified in the context of the Security Target, AGD documentation, and any supplementary information required by the cPP rather than as a separate list specifically for the purposes of CC evaluation. The direct identification of documentation requirements and their assessment as part of the Evaluation Activities for each SFR also means that the tracing required in ADV_FSP.1.2D is treated as implicit, and no separate mapping information is required for this element.

However, if the evaluator is unable to perform some other required Evaluation Activity because there is insufficient design and interface information, then the evaluator is entitled to conclude that an adequate functional specification has not been provided, and hence that the verdict for the ADV_FSP.1 assurance component is a 'fail'.

[13] The Security Target and AGD documentation are public documents. Supplementary information may be public or proprietary: the cPP and/or Evaluation Activity descriptions will identify where such supplementary documentation is permitted to be proprietary and non-public.s

6.2.1.3 AGD: Guidance Documentation

The guidance documents will be provided with the ST. Guidance must include a description of how the IT personnel verifies that the Operational Environment can fulfill its role for the security functionality. The documentation should be in an informal style and readable by the IT personnel.

Guidance must be provided for every operational environment that the product supports as claimed in the ST. This guidance includes:

- instructions to successfully install the TSF in that environment; and
- instructions to manage the security of the TSF as a product and as a component of the larger operational environment; and
- instructions to provide a protected administrative capability.

Guidance pertaining to particular security functionality must also be provided; requirements on such guidance are contained in the Evaluation Activities specified in the SD.

6.2.1.3.1 Operational User Guidance (AGD_OPE.1)

The operational user guidance does not have to be contained in a single document. Guidance to users, administrators and application developers can be spread among documents or web pages.

The developer should review the Evaluation Activities contained in the SD to ascertain the specifics of the guidance that the evaluator will be checking for. This will provide the necessary information for the preparation of acceptable guidance.

Evaluation Activities

The evaluator shall check the requirements below are met by the guidance documentation.

Guidance documentation shall be distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

Guidance documentation must be provided for every Operational Environment that the product supports as claimed in the Security Target and must adequately address all platforms claimed for the TOE in the Security Target.

The contents of the guidance documentation will be verified by the Evaluation Activities defined below and as appropriate for each individual SFR in section 2 above.

In addition to SFR-related Evaluation Activities, the following information is also required.

a) The guidance documentation shall contain instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

b) The documentation must describe the process for verifying updates to the TOE by verifying a digital signature. The evaluator shall verify that this process includes the following steps:

1) Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).

2) Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature.

c) The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The guidance documentation shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.

6.2.1.3.2 Preparative Procedures (AGD_PRE.1)

As with the operational guidance, the developer should look to the Evaluation Activities to determine the required content with respect to preparative procedures.

Evaluation Activities

The evaluator shall check the requirements below are met by the preparative procedures.

The contents of the preparative procedures will be verified by the Evaluation Activities defined below and as appropriate for each individual SFR in section 2 above.

Preparative procedures shall be distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

The contents of the preparative procedures will be verified by the Evaluation Activities defined below and as appropriate for each individual SFR in section 2 above.

In addition to SFR-related Evaluation Activities, the following information is also required.

Preparative procedures must include a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target). The documentation should be in an informal style and should be written with sufficient detail and explanation that they can be understood and used by the target audience (which will typically include IT staff who have general IT experience but not necessarily experience with the TOE product itself).

Preparative procedures must be provided for every Operational Environment that the product supports as claimed in the Security Target and must adequately address all platforms claimed for the TOE in the Security Target.

The preparative procedures must include

a) instructions to successfully install the TSF in each Operational Environment; and

b) instructions to manage the security of the TSF as a product and as a component of the larger operational environment; and

c) instructions to provide a protected administrative capability.

6.2.1.4 Class ALC: Life-cycle Support

At the assurance level provided for TOEs conformant to this cPP, life-cycle support is limited to end-user-visible aspects of the life-cycle, rather than an examination of the TOE vendor's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it is a reflection on the information to be made available for evaluation at this assurance level.

6.2.1.4.1 Labelling of the TOE (ALC_CMC.1)

This component is targeted at identifying the TOE such that it can be distinguished from other products or versions from the same vendor and can be easily specified when being procured by an end user. A label could consist of a "hard label" (e.g., stamped into the metal, paper label) or a "soft label" (e.g., electronically presented when queried).

The evaluator performs the CEM work units associated with ALC_CMC.1.

6.2.1.4.2 TOE CM Coverage (ALC_CMS.1)

Given the scope of the TOE and its associated evaluation evidence requirements, the evaluator performs the CEM work units associated with ALC_CMS.1.

6.2.1.5 Class ATE: Tests

Testing is specified for functional aspects of the system as well as aspects that take advantage of design or implementation weaknesses. The former is done through the ATE_IND family, while the latter is through the AVA_VAN family. For this cPP, testing is based on advertised functionality and interfaces with dependency on the availability of design information. One of the primary outputs of the evaluation process is the test report as specified in the following requirements.

6.2.1.5.1 Independent Testing – Conformance (ATE_IND.1)

Testing is performed to confirm the functionality described in the TSS as well as the guidance documentation (includes “evaluated configuration” instructions). The focus of the testing is to confirm that the requirements specified in Section 5 are being met. The Evaluation Activities in the SD identify the specific testing activities necessary to verify compliance with the SFRs. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/TOE combinations that are claiming conformance to this cPP.

Evaluation Activities

The evaluator shall examine the TOE to determine that the test configuration is consistent with the configuration under evaluation as specified in the ST.

The evaluator shall examine the TOE to determine that it has been installed properly and is in a known state.

The evaluator shall prepare a test plan that covers all of the testing actions for ATE_IND.1 in the CEM and in the SFR-related Evaluation Activities. While it is not necessary to have one test case per test listed in an Evaluation Activity, the evaluator must show in the test plan that each applicable testing requirement in the SFR-related Evaluation Activities is covered.

The test plan identifies the platforms to be tested, and for any platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.

The test plan describes the composition and configuration of each platform to be tested, and any setup actions that are necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of any cryptographic engine to be used (e.g. for cryptographic protocols being evaluated).

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives, and the expected results.

The test report (which could just be an updated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure, so that a

fix was then installed and then a successful re-run of the test was carried out, then the report would show a “fail” result followed by a“pass” result (and the supporting details), and not just the “pass” result[14].

6.2.1.6 Class AVA: Vulnerability Assessment

For the first generation of this cPP, the iTC is expected to survey open sources to discover what vulnerabilities have been discovered in these types of products and provide that content into the AVA_VAN discussion. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. This information will be used in the development of future protection profiles.

6.2.1.6.1 Vulnerability Survey (AVA_VAN.1)

Appendix A in [SD] provides a guide to the evaluator in performing a vulnerability analysis.

Evaluation Activities

The evaluator shall document their analysis and testing of potential vulnerabilities with respect to this requirement. This report could be included as part of the test report for ATE_IND, or could be a separate document.

The evaluator formulates hypotheses in accordance with process defined in Appendix A. The evaluator documents the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.5. The evaluator shall then perform vulnerability analysis in accordance with Appendix A.4. The results of the analysis shall be documented in the report according to Appendix A.5.

[14] It is not necessary to capture failures that were due to errors on the part of the tester or test environment. The intention here is to make absolutely clear when a planned test resulted in a change being required to the originally specified test configuration in the test plan, to the evaluated configuration identified in the ST and guidance documentation, or to the TOE itself.

7. TOE Summary Specification

This section provides evaluators and potential consumers of the TOE with a high-level description of each SFR, thereby enabling them to gain a general understanding of how the TOE is implemented. These descriptions are intentionally not overly detailed, thereby disclosing no proprietary information. These sections refer to SFRs defined in Section 6, Security Requirements.

The TOE consists of the following Security Functions:

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

7.1 Security Audit

7.1.1 Audit Data Generation

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, as described in Section 7.1.2. 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 3: Auditable Events, as well as start-up and shut-down of the audit functions.

The syslog daemon will automatically record the date and time (accurate to the second) for each event.

The TSF has two primary methods of storing and rotating audit records. In the first method which we will call “Method 1”, 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 rotation functionality of these audit records are described in Section 7.1.2 below.

In the second method which we will call “Method 2”, the TSF stores protocol specific audit event records in the /var partition of the underlying filesystem. The storage location and rotation functionality of these audit records are described in Section 7.1.2 below. ‘Method 2’ audit records contain the following event categories and their associated locations within the TOE’s filesystem:

- TLS client errors: /var/log/syslog and /var/log/remote-syslog.
- TLS server errors: /var/log/nginx/nginx-error.log .
- SSH TLS errors: /var/log/auth.log.
- TLS certificate import errors: /var/log/nginx/error.log

To view the audit records in the bulleted list above, the administrative user must login as root via the local console and navigate to those directories to view the logs.

For all audit records (i.e. Method 1 and Method 2), 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].

The TSF has GUI wrapper code, which captures each administrative action and generates the appropriate audit logs. This wrapper is aware of the current user identity and includes it with each of these auditable event types.

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.

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].

The TOE generates audit records for all of the events defined for FAU_GEN.1.1 listed in Section 6.1.1.1, including the SFR-specific auditable events listed in Table 3, Section 6.1.1.1. The audit records generated for the required events are distributed/stored between Method 1 and Method 2 repositories.

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.

Only successfully identified and authenticated administrators can view/read, modify, or delete logs on the TOE.

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.1.1

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 3.

FAU_GEN.1.2; FAU_GEN.2.1

7.1.2 Audit Storage

The TSF secures the transmission of audit records to the remote audit server using syslog over TLS with syslog-ng and OpenSSL.

For “Method 1” described in Section 7.1.1 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 explained in Section 7.1.1. The database is 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 170GB). 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).

FAU_STG_EXT.1.3(Method1)

For ‘Method 2” described in Section 7.1.1 above, the TSF rotates audit records so that at a minimum, 9GB of storage space is maintained on the /var partition of the filesystem. Note that the total physical storage space differs between models of the TOE; however, every model will maintain at least 9GB on the /var partition by performing this audit record rotation functionality. The TSF checks the /var partition every hour, and if current storage utilization reaches 90% of the

allocated partition size, this repository begins rotating audit records. The rotation function deletes the oldest batched/archived audit records (usually each batch contains 15KB of audit records). The function will continue to delete batched/archived audit records until the /var partition is at or below 90% utilization.

FAU_STG_EXT.1.3(Method2)

Audit records are protected from unauthorized access by the restrictive web GUI and 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/modify/delete locally stored audit records.

FAU_STG.1.1; FAU_STG.1.2

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 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.

FAU_STG_EXT.1.1; FAU_STG_EXT.1.2; FAU_STG_EXT.1.3

7.2 Cryptographic Support

The TSF utilizes OpenSSL to implement cryptographic operations. Table 5 below lists the cryptographic algorithms implemented in the TOE, the TSF for which they are utilized, and their corresponding CAVP certificate number:

Table 5: Cryptographic Algorithms

SFR	Scheme Description	Purpose / Service	Cert
FCS_CKM.1	RSA 2048-bit Key-Pair Generation	[+] RSA Key-gen for TLS (Server) and SSH authentication [+] RSA key-transport for all TLS sessions utilizing TLS_RSA_-based ciphersuites	C645
	Diffie-Hellman-Group-14-SHA1 and diffie-hellman-group14-sha1 key-pair generation (2048-bit key)	[+] Key-pair generation for cipher key-establishment during SSH sessions that use the diffie-hellman-group-14-sha1 key-exchange algorithm	N/A*
	ECDSA Key-Pair Generation with P-256, P-384, P-521 (SSH only) key-sizes	[+] Key-pair generation for cipher key-establishment during SSH and TLS	C645

		(Client/Server) sessions that use ECDH-based algorithms. [+] ECDSA Key-gen for TLS Server authentication	
FCS_CKM.2	Elliptic Curve Diffie-Hellman SP800-56A Key Establishment	[+] Key-establishment during SSH sessions that utilize ECDH-based key-exchange algorithms [+] Key-establishment during TLS sessions that utilize ECDHE-based ciphersuites	C645
	Diffie-Hellman-Group-14-SHA1 2048-bit Modulus **NOTE**: the TOE implements RFC 3526 Section 3	[+] Key-establishment during SSH sessions that utilize the diffie-hellman-group-14-sha1 key-exchange algorithm	N/A
FCS_COP.1.1/Data Encryption	Encryption & Decryption using AES with 128 & 256-bit key-sizes in CBC, CTR & GCM modes	[+] Encryption & Decryption in SSH sessions [+] Encryption & Decryption in TLS sessions	C645
FCS_COP.1.1/SigGen	RSA 2048-bit Signature Generation, Signature Verification with SHA-1/256/384/512 Hash	[+] Sig_gen & Sig_ver of ssh-rsa host-keys for SSH connections [+]Sig_gen & Sig_ver of RSA-based x509 certificates for TLS connections	C645
FCS_COP.1.1/Hash	SHA-1/256/384/512	[+] Digest for use in signature generation and verification [+] data integrity and MAC for TLS and SSH connections [+] Obfuscation of locally stored user passwords [+] Use of SHAs in NIST-P curves for ECDH-based key-exchange algorithms for SSH [+] Use of SHAs in NIST-P curves for ECDHE-based ciphersuites for TLS	C645
FCS_COP.1.1/KeyedHash	hmac-sha-1, hmac-sha2-/256/384/512	[+] HMAC in SSH sessions [+] HMAC in TLS Sessions	C645

FCS_RBG_EXT.1	CTR_DRBG using AES 256	[+] Deterministic Random Bit Generation services for cryptographic operations performed for SSH and TLS protected channels	C645

** TD0235 and TD0291 specify the extent of algorithm testing for DH-Group14-sha1. No CAVP certificate is required to satisfy testing of DH-Group14-SHA1 for both Key-generation and Key-establishment. During Functional Testing, the TOE was tested successfully against a known good implementation for the use of DH-Group14-SHA1 in SSH, satisfying the algorithm testing requirements for DH-Group14-SHA1.*

7.2.1 Cryptographic Key Generation and Destruction

The TOE fulfills all of the NIST SP 800-56A and SP 800-56B 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.”

The TOE utilizes OpenSSL for the generation of asymmetric keys. Asymmetric keys are generated for SSH public-key authentication (for both Client and Server), x.509v3 certificates for authentication of the HTTPS web GUI TSF, SSH key-exchange when using diffie-hellman-group-14-sha1 and ECDH-based key-exchange algorithms and for generating x.509v3 certificates for TLS Server authentication. The following asymmetric keys and key sizes are generated by the TOE:

- For SSH Client & Server TSF:
  - Authentication:
    - ssh-rsa key pairs
      - 2048-bits
  - Key-Exchange
    - DH-Group-14-SHA1 key pairs
      - 2048-bits
    - ECDH key-pairs
      - ecdh-sha2-nistp256
        - P-256
      - ecdh-sha2-nistp384
        - P-384
      - ecdh-sha2-nistp521
        - P-521
- For TLS Server TSF:
  - RSA key-pairs:
    - 2048-bits
  - ECDSA key-pairs:
    - NIST P-curves:
      - P-256

FCS_CKM.1.1

The TOE utilizes cryptographic key-establishment schemes when negotiating an SSH Server Trusted Path, TLS Trusted Channels, and TLS Trusted Paths. The following list provides the key-establishment schemes utilized and the purpose of their use:

- 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
    - When RSA-key-transport-based ciphersuites are negotiated/used during TLS session negotiation to the TLS Server TSF and TLS sessions to a remote audit server
        - TOE acts as a sender and recipient
- Elliptic curve-based key-establishment schemes meeting NIST SP 800-56A Rev2
    - When ECDHE-based ciphersuites are negotiated/used during TLS session negotiation to the TLS Server TSF and TLS sessions to a remote audit server
        - TOE acts as a sender and recipient
    - When ECDH-based key-establishment algorithms are negotiated/utilized during SSH Server session negotiation
        - TOE acts as a sender and recipient
- Diffie-Hellman-Group-14-SHA1 meeting Section 3 of RFC 3526
    - When Diffie-Hellman-Group-14-SHA1 key-exchange is negotiated/used during SSH Server session negotiation
        - TOE acts as a sender and recipient

For Diffie-Hellman-Group-14-SHA1, the TOE implements the 2048-bit MODP group as defined in RFC 3526, Section 3.

FCS_CKM.2.1

The table below describes each of the secret keys, private keys, and CSPs used to generate keys.

Table 6: Cryptographic CSPs

CSP Name & Library	Description	Storage
RSA SGK - OpenSSL	RSA (2048 bits) Signature Generation Key	Volatile memory (RAM)
AES EDK - OpenSSL	AES (128/256) Encrypt/Decrypt Key	Volatile memory (RAM)
HMAC Key - OpenSSL	Keyed hash key (160, 256, 384, 512)	Volatile memory (RAM)
DH Private - OpenSSL	DH (Diffie-Hellman) private key agreement key	Volatile memory (RAM)
ECDH Private - OpenSSL	ECDH (P-256, P-384, and P-521) private key agreement key	Volatile memory (RAM)
RNG CSPs - OpenSSL	Entropy input (256 bits), personalization string (128 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 key for OpenSSH authentication	Volatile memory (RAM), Local Filesystem (SSD)
Server Private Keys - TLS	Private RSA 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 7: Cryptographic Public Keys		
Public Key Name & Library	Description	Storage
RSA SVK - OpenSSL	RSA (2048 bits) Signature Verification Key	Volatile memory (RAM)
RSA KEK - OpenSSL	RSA (2048 bits) Key Encryption (public key transport) Key	Volatile memory (RAM)
DH Public - OpenSSL	DH (Diffie-Hellman) private key agreement key	Volatile memory (RAM)
ECDH Public - OpenSSL	ECDH (P-256, P-384, and P-521) public key agreement key	Volatile memory (RAM)

The TSF zeroizes volatile secret and private keys when power is removed[15]. 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.

Persistent cryptographic keys are generated by the administrator in the following scenarios:

- when an SSH host-key is needed for the SSH Server TSF,
- when creating a Certificate Signing Request (and subsequent importing of the CA-signed certificate) for the TLS Server TSF,
- and when importing trusted Certificate Authorities (x.509) into the TSF's certificate trust stores.

These persistent cryptographic keys are stored in the underlying filesystem of the TOE (non-volatile storage).

There are three scenarios in which persistent cryptographic keys stored in non-volatile storage are securely destroyed.

- Persistent keys in non-volatile storage can be zeroized by the Security Administrator performing a secure erase procedure. "Secure erase" utilizes the secure erase command provided by the SSD vendor, as run via a script (by the Security Administrator) that directly sends the command to the SSD. There are two possible compliant secure erase behaviors that this initiates in the drive.
    - Scenario One: First, the drive may physically delete each 'erase block' (an erase block is the minimum physical storage size that an SSD drive can erase in a single erase operation on the SSD) by effectively writing over the storage space with zeros (i.e. logically addresses the storage location of the key and performs a single overwrite consisting of zeros).
    - Scenario Two: The drive may generate a new key to use in writing to the drive, overwriting the old key (i.e. logically addresses the storage location of the key and performs a single overwrite consisting of a new value of the key). This causes any data on the drive to become inaccessible because the key to decrypt it is destroyed. All data on externally connected drives are encrypted with AES-256. Therefore, zeroizing these keys will cause the loss of all user data.
- Scenario Three: The last instance where persistent keys may be destroyed is when an administrative user deletes the associated public-key via the administrative web GUI. Zeroization of persistent keys in non-volatile storage is done by logically addressing the

[15] This method of zeroization meets the NSA CSS Storage Device Declassification Manual for the zeroization of DRAM and SRAM.

storage location of the key and performing a single overwrite of the key-data, with a new value of the key (i.e. logically addresses the storage location of the key and performs a single overwrite consisting of a new value of the key). OpenSSL provides this zeroization service.

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. In addition, the reference to the key is deleted, and a request for garbage collection directly follows.

FCS_CKM.4.1

7.2.2 Cryptographic Operations

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 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.

The TOE provides RSA and ECDSA digital signatures (signature generation and signature verification) for ECDSA (TLS only) and RSA-based cryptographic functions performed in both TLS Server/Client and SSH Server TSF. The implementation of RSA key generation for these security functions conform to FIPS Pub 186-4 Appendix B.3. RSA key size of 2048-bits is supported for both TLS (server and client) and SSH server security functionality. 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 size of P-256. RSA key size of 2048-bits is supported for both TLS (server and client). 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. This is enforced by the OpenSSL cryptographic module and is not configurable by the security administrator.

The TOE performs SHA-1, 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, and HMACs and Key Derivation Functions in TLS Client and Server TSF. The TOE generates hashes with 160, 256, 384, or 512-bit message digest size.

The TOE performs keyed-hash message authentication using hmac-sha1, hmac-sha2-256, hmac-sha2-384, or hmac-sha2-512 and using cryptographic key sizes of 160, 256, 384, 512-bits and message digest sizes of 160, 256, 384, or 512-bits, with block sizes of 512-bits for SHA-1 and SHA-256, and 1024-bits for SHA-384 and SHA-512, per RFC 2104 and RFC4868.

FCS_COP.1/DataEncryption; FCS_COP.1/SigGen; FCS_COP.1/Hash; FCS_COP.1/KeyedHash

7.2.3 HTTPS Protocol

The TOE provides an HTTPS protected administrative web GUI for remote management of the TOE, supporting TLSv1.2. The TSF implements the server side of the HTTPS protocol according to RFC 2818. The TSF listens on port 443 for the HTTPS connections through the use of the

Nginx web server platform. The TSF uses HTML over HTTPS to present the administrative users with a secure management interface. The TSF uses TLS to provide a secure connection between the TSF and the administrator; however, HTTP is used to maintain the administrator's session. The underlying HTTP server performs authentication by passing the username/password credentials to the Linux PAM library. TLS client certificate authentication is not supported.

This interface is accessed using the HTTPS's scheme 'https:' URI.

FCS_HTTPS_EXT.1.1; FCS_HTTPS_EXT.1.2

The TSF does not require/request client certificate-based authentication.

FCS_HTTPS_EXT.1.3

7.2.4 Random Bit Generation

The TOE utilizes the CTR_DRBG (AES 256) mode as specified in NIST SP800-90A for TLS and SSH cryptographic functions. The primitives used include the values of 'V' and 'Key', which are described in NIST SP800-90A Section 10.2.1.1. The security strength is equal to the AES key size (256-bits). This DRBG is seeded with at least 256-bits of entropy [16] to initialize the DRBG.

Pure Storage's implementation calls RDSEED, the output of the hardware-based noise source built into the Intel CPUs utilized by the TOE, also known as Intel® Secure Key, from a single thread. Since every call to RDSEED is XOR'd with the output of RDRAND, RDRAND is reseeded every four invocations of RDSEED. The implementation calls RDSEED four times and ensures to check if the call to RDSEED was successful since RDSEED is non-blocking. The output of RDSEED is directly used to initialize OpenSSL's CTR_DRBG (AES).

FCS_RBG_EXT.1.1; FCS_RBG_EXT.1.2

7.2.5 SSH Server Protocol

The TSF uses OpenSSH 7.2p2-4ubuntu2.8 for its SSH implementation, which was compiled to rely on OpenSSL for all cryptographic operations. This version of OpenSSH supports the RFCs 4251-4254, 5656, 8332 and 6668.

FCS_SSHS_EXT.1.1

The TSF will validate an SSH packet length field (by comparing the length field value to the actual bytes received; process of counting bytes received is described below) that the TOE received and terminate the connection if the SSH packet received is greater than 262,144 bytes in length. OpenSSH calculates the size of the incoming SSH packets. The size calculated is of the entirety of the SSH application data as handed to the SSH server process after processing by the lower layers of the network stack. Once de-capsulated, the SSH application data is handed to the socket that was established for the SSH Server process. The SSH application performs the calculation of this data that it is handed. If the SSH application data is calculated to be greater than 262,144 bytes, the connection is terminated and an audit record is generated.

FCS_SSHS_EXT.1.3

[16] Vendor assumption of the amount of min-entropy provided by the entropy source is 4-bits of min-entropy per byte of raw data from the entropy source. Entropy source is third-party provided. The DRBG is provided 512-bits of raw data from the entropy source when being initialized. The assumption is that at least 256-bits of min-entropy is being provided to the DRBG for initialization.

The SSH Server supports public key-based and password-based authentication as detailed in RFC 4252. The following public-key algorithms are acceptable for use for authentication and are not configurable by the security administrator:

- ssh-rsa
- rsa-sha2-256
- rsa-sha2-512

FCS_SSHS_EXT.1.2; FCS_SSHS_EXT.1.5

The SSH Server TSF only supports the following encryption algorithms:

- aes128-cbc
- aes256-cbc
- aes128-ctr
- aes256-ctr
- aes128-gcm@openssh.com
- aes256-gcm@openssh.com

All other encryption algorithms are rejected by the TSF. This behavior is not configurable by the security administrator.

FCS_SSHS_EXT.1.4

The TOE's Server implementation supports only the following MAC algorithms:

- hmac-sha1
- hmac-sha2-256
- hmac-sha2-512
- "Implicit"

The TSF rejects all other MAC algorithms for the SSH Server transport implementation.

When aes128-gcm@openssh.com or aes256-gcm@openssh.com is negotiated as the encryption algorithm in SSH sessions with the TOE, the MAC algorithm field is ignored and GCM is implicitly used as the data integrity MAC for that session. "implicit" is not an SSH algorithm identifier and will not be seen on the wire as such; however, the negotiated MAC might be decoded as "implicit", thus this identifier being used in this Security Target.

FCS_SSHS_EXT.1.6

The TOE's SSH Server implementation allows only the following key-exchange algorithms:

- diffie-hellman-group14-sha1
- ecdh-sha2-nistp256
- ecdh-sha2-nistp384
- ecdh-sha2-nistp521

FCS_SSHS_EXT.1.7

The TSF ensures that SSH Server connections are re-keyed after no more than one (1) hour, or no more than (1) gigabyte of SSH encrypted data has been transmitted, while using the same key. OpenSSH starts a timer when an SSH session is established. Once the timer reaches one (1) hour, the SSH Client forces a renegotiation of the key-exchange phase of an SSH session establishment, which ultimately leads to the creation of a new session key. If the rekeying step is not successfully negotiated, the SSH session is terminated. If one (1) gigabyte of encrypted SSH data is transmitted, the TSF will initiate a rekey on the next SSH data received or transmitted. Rekeying is performed upon exceeding the threshold that is reached first.

FCS_SSHS_EXT.1.8

Note that all SSH cryptographic settings are hardcoded into the TOE and not configurable by the administrator.

7.2.6 TLS Client Protocol

The TOE utilizes OpenSSL 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 ciphersuites are supported:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_ SHA256
- 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
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

FCS_TLSC_EXT.1.1

Reference identifiers are created by the TOE using the configuration data provided by the admin in administrative web GUI. 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. The transport protocol is also chosen (TLS in the evaluated configuration). 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.

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.

The TSF supports the following reference identifiers:

- CN-ID
- DNS-ID
- IPv4 address in the SAN

The TSF establishes reference identifiers for the remote audit server as follows:

- 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.
- 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.

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). A match occurs if the reference identity octet string and value octet strings are identical.

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:

- in step 1, the domain name is considered a "stored string";
- in step 3, the flag called "UseSTD3ASCIIRules" is set;
- in step 4, each label is processed with the "ToASCII" operation; and
- in step 5, all label separators are changed to U+002E (full stop/period).

Canonical format (RFC 3986 for IPv4) is not enforced by the TSF.

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 Section 7.3.6.

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.

The TOE generates EC Diffie-Hellman parameters over the following NIST curves:

- secp256r1
- secp384r1

This behavior is performed by default and is not configurable by the security administrator.

FCS_TLSC_EXT.1.2; FCS_TLSC_EXT.1.3; FCS_TLSC_EXT.1.4

7.2.7 TLS Server Protocol

As described in Section 7.2.3, the TOE provides a TLS protected web GUI administrative interface for remote management of the TOE. TLS is implemented by the OpenSSL module. The TOE supports TLSv1.2, rejecting all other SSL/TLS versions. The following ciphersuites are supported:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256
- TLS_ECDHE_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
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

FCS_TLSS_EXT.1.1; FCS_TLSS_EXT.1.2

The TOE generates EC Diffie-Hellman parameters over the following NIST curves:

- secp256r1

This behavior is performed by default and is not configurable by the security administrator.

The Server Key Exchange message is used to convey the server's ephemeral Diffie-Hellman and ECDH public key (and the corresponding elliptic curve domain parameters) to the client. It is sent by the server only when the server Certificate message does not contain enough data to allow the client to exchange a premaster secret. This occurs when the following key exchange methods are negotiated:

- TLS_DHE_RSA_*
- TLS_ECDHE_RSA_*
- TLS_ECDHE_ECDSA_*

When ephemeral ECDH is negotiated, the Server Key Exchange message contains the following parameters:

- Curve Type parameter
  - named_curve
    - Identifies that a named curve is the method of identifying the curve used in the server key exchange message
- Named Curve parameter
  - Identifies the actual named elliptic curve that is used by the server in the server key exchange message.
- Curve Params parameter
  - Specifies the elliptic curve domain parameters associated with the ECDH public key
- Pubkey
  - Contains the actual ephemeral public key data for the ECDH key exchange
- Signature Algorithm
  - Identifies the signature algorithm utilized by the server to sign the key exchange message
- Signature Hash Algorithm
  - Identifies the hash algorithm utilized by the server in signing the key exchange message
- Signed Params
  - A hash of the server key exchange parameters, the client hello random, and server hello random data, with the signature appropriate to that hash applied. The private key corresponding to the certified public key in the server's Certificate message is used for signing. Signature algorithm is RSA for TLS_ECDHE_RSA_*. Signature algorithms is ECDSA for TLS_ECDHE_ECDSA_*.

When ephemeral DH is negotiated, the Server Key Exchange message contains the following parameters:

- dh_p parameter
  - The prime modulus used for the Diffie-Hellman operation

- dh_g
  - The generator used for the Diffie-Hellman operation
- dh_Ys
  - The server's Diffie-Hellman public value (g^X mod p)
- Signed Params
  - A signature of the client random, server random and server's key exchange parameters, using the end entity server certificate sent in the server's certificate message

The TOE supports the secp256r1 elliptic curves. The TOE supports only the uncompressed format for these curves, in the Key Exchange Message.

The TOE generates Diffie-Hellman parameters of size 2048-bits and presents them in the server key_exchange message.

FCS_TLSS_EXT.1.3

7.3 Identification and Authentication

7.3.1 Authentication Failure Management

The TSF can be administered through three interfaces, the local console, HTTPS/TLS, and SSH.

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.

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 (SSH_RSA). 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 SSH_RSA authentication or username/password match an authorized administrator's credentials, the user is granted access to the command line interface.

The web GUI interface is protected by the HTTPS/TLS Trusted Path. Administrative users navigate to *http://<management_ip_address>*, utilizing one of the supported web browsers listed in Section 1.3.4 to access the interface. This will redirect the user to the HTTPS protected interface hosted on TCP port 443. Navigating to *https://<management_ip_address>* will bring the user directly to the protected web GUI interface. When a user connects to the HTTPS/TLS interface, the user is initially presented with the configured Warning and Consent banner, which the user must accept prior to continuing to the username/password authentication form. When a user submits a username/password combination, the TSF attempts to authenticate the user. If the username/password match an authorized administrator's credentials, the user is granted access to web-based graphical user interface.

For all three 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.

The TSF supports passwords that include any character that can be entered from a standard US keyboard: upper and lower case letters, numbers, and ASCII hexadecimal codes 0x20-0x2F, 0x3A-0x40, 0x5B-0x60, 0x7B-0x7E.

For each login method, a customizable banner is displayed immediately before the username/password prompt or username/public key exchange.

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.

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. Failed authentication attempts is 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. To reiterate, the 'pureuser' account will always be accessible at the local console. This account can be used to unlock any locked account.

FIA_AFL.1

7.3.2 Password Management

Passwords created for user authentication to the TOE's local and remote administrative interfaces may be composed of the following:

- 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

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.

FIA_PMG_EXT.1

7.3.3 User Identification and Authentication

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

- View the warning and consent banner in accordance with FTA_TAB.1
- Respond to ICMP Echo Request
- Respond to ARP requests with ARP replies
- Make DNS Requests
- Respond to HTTP Get Requests on TCP port 80 with an HTTP 301 'Moved Permanently' Status Code redirecting to TCP port 443
- Respond to TLS Client_Hello messages with TLS Server_Hello messages on TCP port 443

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.

Section 7.3.1 of this document describes the logon process for each logon method (local, remote (HTTPS, SSH)) supported for the product. A successfully authenticated administrative user will be presented with a management interface (SSH CLI or Web GUI).

FIA_UIA_EXT.1

7.3.4 Password-based Authentication Mechanism

The TOE provides a local password-based authentication mechanism to perform local administrative user authentication. This functionality is provided by the Linux PAM module. Users' passwords are stored as a SHA-512 hash (including a salt) in the underlying filesystem. There are no other authentication mechanisms to perform local administrative user authentication.

FIA_UAU_EXT.2

7.3.5 Protected Authentication Feedback

At the local console, the TOE does not echo back the characters typed in for the password credential. Using the administrative web GUI, the characters are echoed back as dots, effectively obscuring the password from shoulder-surfing.

FIA_UAU.7

7.3.6 X.509 Certificate Validation

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.

Certificates are checked for validity when the TOE's HTTPs/TLS Server certificate is presented to a TLS client, such as when an administrative user is connecting to the administrative web GUI. The TOE will validate its own TLS Server certificate in these instances, prior to sending the certificate to the peer for authentication. In addition, when 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.

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:

1. The current date is between the "Valid from" and "Valid to" dates listed in the certificate
2. The certificate is not revoked when a response from an OCSP responder is successfully provided to the TSF
3. The certificate chain is valid:
    a. Each certificate in the certificate chain passes the checks described in #1 and #2 above.
    b. Each certificate (other than the first certificate) in the certificate chain has the basicConstraints extension 'Subject Type=CA'.
    c. Each certificate is signed by:
        i. a certificate (which has the 'certificate signing' key-usage extension) in the certificate chain installed on the TOE, or

ii. a trusted root CA that has been installed on the TOE

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):

- id-kp-serverAuth OBJECT IDENTIFIER ::= { id-kp 1 }
- id-kp-clientAuth OBJECT IDENTIFIER ::= { id-kp 2 }
- id-kp-OCSPSigning OBJECT IDENTIFIER ::= { id-kp 9 }

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 ignore all other key extended usage purposes.

While the TOE recognizes the id-kp-clientAuth purpose, there is no TSF for which this purpose would be used. In the case where the TOE is the TLS server (web GUI remote administrative interface), the TSF requires that the TOE's TLS server certificate contains the id-kp-serverAuth purpose before presenting this certificate to the TLS client.

The TOE supports a minimum certificate path length of (3) three, thus at minimum a Root CA, two intermediate CAs, and an end-entity/leaf certificate are supported.

FIA_X509_EXT.1/Rev

7.3.7 X.509 Certificate Authentication

The administrative web GUI 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 two intermediate CA's.

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. The configured/installed ECDSA certificate is sent to the peer when an ECDSA-authentication based ciphersuite is chosen by the TOE during a TLS handshake with a TLS client.

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.

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.2

7.3.8 X.509 Certificate Requests

CSRs are only generated for the TLS Server TSF. Administrators are able to generate a CSR through the local and remote CLI and via the administrative web GUI management 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:

- Common Name
- Organization

- Organizational Unit
- Country

FIA_X509_EXT.3

7.4 Security Management

The TOE does not provide any unauthenticated services other than the access banners provided at each login prompt.

The TOE provides the ability to administer the TOE remotely through SSH and HTTPS/TLS. For HTTPS, a custom, restrictive GUI is provided through a custom web application running on Jetty 9.4.6.v20170531 and Nginx nginx/1.4.6. For SSH and the local console, the user is presented with a restrictive CLI provided through rbash that restricts the user to a specific list of commands that do not allow for general computing or reading of private keys. For SSH, OpenSSH is used to handle the trusted path.

The TSF includes four administrative roles within the Authorized Administrator role: Internal Administrator, Array Administrator, Storage Administrator, and Read-Only Administrator. The TOE includes three default accounts: pureuser, os76, and root. pureuser is the customer-facing default account under the Array Administrator role. os76 and root are hard-coded accounts, which are only accessible for update and support issues and are configured to be disabled otherwise in the CC-evaluated configuration. Users are locally authenticated and are defined by as either an Array Administrator, Storage Administrator, or Read-Only Administrator.

7.4.1 Management of Security Functions Behaviour

The TOE restricts the ability to determine and modify the behavior of the transmission of audit data to an external IT entity to 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.

The TOE in the evaluated configuration only supports TCP for the TLS Trusted Channel to the remote audit server, so the administrative guidance documentation instructs the admin to only configure the channel to utilize TCP. Only identified and authenticated administrators are able to determine the configuration/behavior of the remote auditing configuration.

FMT_MOF.1/Functions

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. The administrative interfaces are only accessible to administrators after successful authentication; consequently, this functionality is not available to non-administrators.

FMT_MOF.1/ManualUpdate

The TSF restricts the ability to start and stop services to Security Administrators. The Security Administrator is able to start/stop the service to send TSF generated audit records to a remote audit server.

FMT_MOF.1/Services

7.4.2 Management of TSF Data

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. 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.

FMT_MTD.1/CoreData

Cryptographic keys can only be manage by identified and authenticated security administrators to the TOE. This functionality is possible by virtue of restricted access to TOE's administrative interfaces. Management of cryptographic keys 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. Administrators of the TOE can generate, import, and delete cryptographic keys via all administrative interfaces.

FMT_MTD.1/CryptoKeys

7.4.3 Specification of Management Functions

Identified and authenticated security administrators have the ability to do the following:

- Ability to administer the TOE locally and remotely;
  - Locally via the local console, remotely via SSH and the web GUI interfaces.
- Ability to configure the access banner;
  - One, global banner is configured, which is presented at the local and remote administrative interfaces (local console and SSH respectively). This global banner can be configured via the local or remote management interface.
- Ability to configure the session inactivity time before session termination or locking;
  - Session inactivity time for the local, remote CLI consoles (local console and SSH), and web GUI is able to be configured via any of those interfaces.
- Ability to update the TOE and to verify the updates using hash comparison capability prior to installing those updates;
  - The update package is verified using hash comparison.
    - This is accessible in the local and SSH remote interfaces.
- Ability to configure the authentication failure parameters for FIA_AFL.1;
  - This is configured in either administrative user interface.
- Ability to start and stop services;
  - This is configured in either administrative interface.
- Ability to configure audit behaviour;
  - This is configured in either administrative interface.
- Ability to manage the cryptographic keys;
  - This is configured in either administrative interface.
- Ability to configure the cryptographic functionality;
  - This is configured in either administrative interface.
- Ability to re-enable an Administrator account;
  - This is configured in either administrative interface.
    - Note: while the TSF for FIA_AFL.1 is a time-based lockout mechanism, administrators have the ability to un-lock offending accounts prior to the lockout time elapsing. Unlocking can be achieved through all administrative interfaces.
- Ability to set the time that is used for time-stamps;

    - This is configured in either administrative interface.
- Ability to configure the reference identifier for the peer;
    - This is configured in either administrative interface.
- Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
    - This is configured in either administrative interface.
- Ability to import X.509v3 certificates to the TOE's trust store;
    - This is configured in either administrative interface.

FMT_SMF.1

7.4.4 Restrictions on Security Roles

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:

- Internal Administrator
- Array Administrator
- Storage Administrator
- Read-Only Administrator

All roles listed above are considered authorized 'Security Administrators.' All administrators may administer the TOE locally and remotely.

FMT_SMR.2

7.5 Protection of the TSF

7.5.1 Protection of Administrator Passwords

Locally stored passwords are stored as SHA-512 hashes (including a salt). The administrative interfaces do not provide methods to view password hashes.

FPT_APW_EXT.1

7.5.2 TSF Testing

Upon power-up, the TSF performs a SHA-1 of the kernel, all executables, and all interpreted files, insuring 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.

FPT_TST_EXT.1

7.5.3 Trusted Update

A first step in establishing trust in the update process is to review the published hash of the candidate update package. The System Administrator can verify the candidate update prior to installing the update, by comparing the computed hash of the candidate update package to the published hash. This establishes that the update downloaded is the one associated with the published hash.

The currently installed version of software can be queried on the TOE by running the '*purearray list'* command from the CLI. In addition, each page of the administrative web GUI contains the current executing software version in the lower right corner of the web page.

Updates are initiated and installed via the internal administrator role. The associated SHA-1 hashes are published in the update release notes. The administrator enables the root account, logs into the TOE as root, uploads the update to the TOE, verifies the hash of the update file using a local OpenSSL command, then proceeds to install the update using procedures provided in the guidance documentation. If the hash verification is unsuccessful, the administrator is instructed to delete the update. Upon completion of the update, guidance instructs the administrator to disable the root account.

FPT_TUD_EXT.1.1; FPT_TUD_EXT.1.3

The TSF provides Security Administrators the ability to manually initiate updates to TOE firmware/software via the local console, SSH, and web GUI remote consoles. There are no other update mechanisms.

There are no delays to the updating of the TOE software. Once the script to install the update packaged is initiated, the TOE performs the installation of the software and the necessary reboot operations to boot using the new software package. 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.

Candidate update packages are available from Pure Storage directories under ftp://ftp.purestorage.com/outgoing/.

The file names of the candidate update package follow the pattern of

- purity_<version>_<build>.ppkg.

FPT_TUD_EXT.1.2

7.5.4 Protection of TSF Data

The GUI is restrictive by design and only allows the administrator access to pre-determined functions. Access to sensitive cryptographic material is not included in those pre-determined functions of the administrative web GUI application. An access control list is implemented in the web GUI application to prevent directory/path traversal to the parts of the file system that contain the symmetric keys and private keys.

The CLI (both local and remote) uses an rbash prompt that restricts users to a specific directory and only allows them access to pre-defined binaries created by Pure Storage. Keys are not stored in a directory accessible via the rbash shell.

FPT_SKP_EXT.1

7.5.5 Reliable Time Stamps

The following TSF security functions utilize the time:

- Audit Record timestamps
- Web GUI session timeout
- SSH Console session timeout
- Local Console session timeout
- x509 Certificate expiration checking
- Input for the 'Random' field in the TLS Client_Hello and Server_Hello Handshake messages, as a 32-bit unsigned integer

The TSF contains a real-time clock to maintain the time between updates from administrator configuration. All TSF security functions use the local real-time clock. The drift of the Real Time Clock (of which the TOE's actual system clock time is based) is expected to be +/-1 second per 30 days. The administrator is instructed to manually update the system time at least once every 6 months to maintain accurate time to support the correct operation of the security functionality performed by the TOE.

FPT_STM_EXT.1

7.6 TOE Access

The TSF allows three methods of administrator access: local serial access and remote SSH and HTTPS/TLS access. 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. The administrator can terminate an administrative session by logging out.

7.6.1 Session Termination

The TSF uses the environment variable TMOUT, set in /etc/profile, in order to enforce user inactivity requirements for the SSH and local console CLI. The TSF uses a Jetty-specific configuration value to enforce user inactivity requirements for the HTTPS GUI.

The TSF allows the user to proactively terminate their session by logging out of the CLI or GUI.

FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4

7.6.2 Default TOE Access Banners

The TSF allows three methods of administrator access: local serial access and remote SSH and HTTPS/TLS access. 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. The administrator can terminate an administrative session by logging out.

FTA_TAB.1

7.7 Trusted Path/Channels

7.7.1 Inter-TSF Trusted Channel

The TSF uses OpenSSL and syslog-ng to communicate with remote audit log servers via TLS.

Each protocol implementation is performed according to the descriptions and requirements provided in Section 7.2.

The TOE initiates the following Trusted Channel communications:

- TLS (as client) channel to remote audit server
  - 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 Section 7.3.6.

FTP_ITC.1

7.7.2 Trusted Path

The TSF uses OpenSSL and OpenSSH to provide a remote CLI interface for administrators, which is protected via SSH. Relevant ciphersuites and algorithms are enumerated in Section 7.2.

The TSF uses Jetty, Nginx, and OpenSSL to provide a remote GUI interface for administrators, which is protected via TLS/HTTPS. Relevant ciphersuites and algorithms are enumerated in Section 7.2.

FTP_TRP.1/Admin

8. Terms and Definitions

Table 8: TOE Abbreviations and Acronyms

Abbreviations/ Acronyms	Description
AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
BIOS	Basic Input/Output System
CA	Certificate Authority
CBC	Cipher Block Chaining
CLI	Command Line Interface
CMOS	Complementary Metal–Oxide–Semiconductor
CRL	Certificate Revocation List
CSP	Critical Security Parameter
CSR	Certificate Signing Request
CTR	Counter
DH	Diffie-Hellman
DHE	Diffie-Hellman Ephemeral
DNS	Domain Name System
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
DTLS	Datagram Transport Layer Security
ECDH	Elliptic Curve Diffie-Hellman
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECDSA	Elliptic Curve Digital Signature Algorithm
EDK	Encrypt/Decrypt Key
ESP	Encapsulating Security Payload
GCM	Galois Counter Mode
GUI	Graphical User Interface
HMAC	Hash Message Authentication Code
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IKE	Internet Key Exchange
IP	Internet Protocol
KAT	Known Answer Test
KEK	Key Encryption Key
KDF	Key Derivation Function
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
PCT	Pairwise Consistency Test
PKCS	Public Key Cryptography Standards
RFC	Requests for Comments
RNG	Random Number Generator
RSA	Rivest-Shamir-Adleman
SA	Security Association
SAN	Subject Alternative Name
SFP	Small Form-Factor Pluggable
SGK	Signature Generation Key
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SPD	Security Policy Database
SSH	Secure Shell

Table 8: TOE Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
SVK	Signature Verification Key
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
UI	User Interface
URI	Uniform Resource Identifier
VPN	Virtual Private Network

Table 9: CC Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
CC	Common Criteria
CCRA	Arrangement on the Recognition of Common Criteria Certificates in the field of IT Security
DOD	Department of Defense
EAL	Evaluation Assurance Level
FIPS	Federal Information Processing Standards
OSP	Organizational Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SFP	Security Function Policy
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification

9. References

Table 10: TOE Guidance Documentation			
Reference	Description	Version	Date
[T1]	Guidance Documentation Pure Storage FlashArray//XR2 and //XR3 Appliances Running Purity 5.3	Version 4.2.5	January, 2021

Table 11: Common Criteria v3.1 References			
Reference	Description	Version	Date
[C1]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model CCMB-2017-04-001	V3.1 R5	April 2017
[C2]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional components CCMB-2017-04-002	V3.1 R5	April 2017
[C3]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components CCMB-2017-04-003	V3.1 R5	April 2017
[C4]	Common Criteria for Information Technology Security Evaluation Evaluation Methodology CCMB-2017-04-004	V3.1 R5	April 2017

Table 12: Supporting Documentation			
Reference	Description	Version	Date
[cPP]	Collaborative Protection Profile for Network Devices	2.1	September 24, 2018
[SD]	Supporting Document Mandatory Technical Document Evaluation Activities for Network Device cPP	2.1	September 2018