Pure Storage FlashArray
Security Target

15-3312-R-0015

Version: 1.1

March 4, 2016

Prepared For:

Pure Storage, Inc.

650 Castro Street, Suite #260

Mountain View, CA 94041

Prepared By:

709 Fiero Lane, Suite 25

San Luis Obispo, CA 93401

Notices:

Table of Contents

Tables

1. Security Target (ST) Introduction

- The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.
- The ST reference shall uniquely identify the ST.
- The TOE reference shall identify the TOE.

The structure of this document is defined by CC v3.1r3 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:	Pure Storage FlashArray Security Target
ST Version Number:	Version 1.1
ST Author(s):	InfoGard Laboratories, Inc.
ST Publication Date:	3/4/2016
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:	Pure Storage FA-405, FA-450, FlashArray//m20, FlashArray//m50, and FlashArray//m70 Series Appliances

1.3 Target of Evaluation Overview

1.3.1 TOE Product Type

The TOE is classified as a Network Device (a generic infrastructure device that can be connected to a network).

1.3.2 TOE Usage

Pure Storage's FlashArray (TOE) is an enterprise Network Attached Storage solution that includes a Linux-based operating system, SAN 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 (FlashReduce)
- Non-Disruptive Expansion and High Availability (FlashProtect)
- Backup & Disaster Recovery (FlashRecover)
- Real-world Optimized Performance (100K - 200K 32K IOPS @ <1ms average latency)
- Data at rest encryption with AES-256

The Pure Storage FlashArray is designed to act as a data storage endpoint for a SAN (Storage Area Network). The TOE supports remote administration over HTTPS/TLS (Hypertext Transfer Protocol Secure/Transport Layer Security) with cryptographic encryption and authentication using FIPS-certified algorithms. The TOE also supports use of external authentication and audit servers, protected using TLS.

1.3.3 TOE Major Security Features Summary

- Audit
- Cryptography
- User Data Protection
- 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

Syslog Server:

- RFC 3164
- TLS Transport Mapping - RFC 5425
    - Required TLS ciphersuites match those required for HTTPS below

NTP Server:

- NTPv4 - RFC 5905

The TOE is known to be compatible with Chrome 47.0 – 48.0 and Firefox 41.0 – 42.0. The TOE requires a Web Browser (Remote Console) supporting:

- Protocol versions (at least one of):

  - HTTPs/TLSv1.1 (RFC 2818 & 3246)
  - HTTPs/TLSv1.2 (RFCs 2818 & 5246)
- Ciphersuites (at least one of):
  - 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

The TOE is known to be compatible with OpenSSH 6.6p1-2ubuntu2. The TOE requires an SSH client (Remote Console) supporting:

- Protocol versions (at least one of):
  - SSHv2 (RFCs 4251-4254, 5656 and 6668)
- Data Encryption (at least one of):
  - AES-CBC-128
  - AES-CBC-256
  - AEAD_AES_128_GCM
  - AEAD_AES_256_GCM
- Data Integrity (at least one of):
  - hmac-sha1
  - hmac-sha1-96
  - hmac-sha2-256
  - hmac-sha2-512
- Key Exchange
  - diffie-hellman-group14-sha1
  - ecdh-sha2-nistp256
  - ecdh-sha2-nistp384
  - ecdh-sha2-nistp521

Active Directory authentication server communicating via LDAP over TLS

The TOE's IT environment must support incoming TCP connections from the PureStorage support staff for trusted updates.

1.3.4.2 Hardware Requirements

Local Console:

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

1 Gigabit Ethernet for Trusted Paths and Trusted Channels

SAS-connected SSD Storage Array from PureStorage

1.4 Target of Evaluation Description

The TOE is a Network Attached Storage device designed for high speed storage with enterprise level protocols and management features.

The TOE consists of one or two physical PCs that are connected together via InfiniBand[1] for high availability purposes. The PCs (TOE) are grouped and sold as five possible models: FA-405, FA-450, //m20, //m50, and //m70. The TOE acts as a SAN storage endpoint over the Fibre Channel and 10GbE (10 Gigabit Ethernet) interfaces, and allows TLS connections to its 1Gb Ethernet management interface.

The TOE operating system, Purity 4.7, is built on the Ubuntu Linux kernel and an Intel Xeon x64 CPU.

1.4.1 Target of Evaluation Physical Boundaries

The TOE consists of the following hardware:

- FA-405
  - PCs: 1x OEM PowerEdge R620
  - CPU: Intel Xeon E5-2640 v2, 8 cores, 2.0 GHz, 30MB Cache
  - RAM: 128 GB DDR3 1600MHz
- FA-450
  - PCs: 2x OEM PowerEdge R720
  - CPU: Intel Xeon E5-2697 v2, 12 cores, 2.7 GHz, 30MB Cache
  - RAM: 512 GB DDR3 1600MHz
- //m20
  - PCs: 1x Custom-built PC
  - CPU: Intel Xeon E5-2630 v3, 8 cores, 2.6 Ghz, 20MB Cache
  - RAM: 192 GB DDR4-1866
- //m50
  - PCs: 2x Custom-built PC
  - CPU: Intel Xeon E5-2670 v3, 12 cores, 2.3 Ghz, 25MB Cache
  - RAM: 256 GB DDR4-2133
- //m70
  - PCs: 2x Custom-built PC
  - CPU: Intel Xeon E5-2698 v3, 16 cores, 2.3 Ghz, 30MB Cache
  - RAM: 512 GB DDR4-2133

Running:

- Purity SW v4.7

The guidance documentation that is part of the TOE is listed in Section 9, “References,” within Table 15: TOE Guidance Documentation.

The TOE has the following types of physical connections:

[1] InfiniBand is a brand of fiber optic interconnectivity solutions. It is a direct connection connecting the two controllers, allow them to operate in sync. There are no security relevant interfaces, and is not available over the network.

- Host IO Cards
  - 10GbE iSCSI
  - 8GB FC
- Management Ports
  - 4x 1GbE
- Replication Ports
  - 4x 10GbE SFP
- SAS Ports
  - 8x SAS3 (12Gb/s) Mini-SAS HD
- USB Ports
  - 4x USB 3.0 Rear
  - 2x USB 2.0 Front

1.4.2 Target of Evaluation Description

The logical boundary of the TOE include those security functions implemented exclusively by the TOE. These security functions are summarized 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 are provided in Section 7, “TOE Summary Specification.”

1.4.2.1 Audit

The TOE audits all events and information defined by the Network Device Protection Profile v1.1. Audit logs 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. Audit events are transmitted to an external IT entity using the TLS protocol. The TOE also protects storage of audit information from unauthorized deletion and modifications.

1.4.2.2 Cryptographic Operations

The TOE implements CAVP validated cryptographic algorithms for random bit generation, encryption/decryption, authentication, and integrity protection/verification. These algorithms are used to provide security for the SSH and TLS protocols.

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

1.4.2.3 User Data Protection

The TOE ensures that any previous information content of network packets are not re-used in subsequent network packets by leveraging the Linux kernel's network packet processing mechanisms. All network resources are zeroized upon allocation of that buffer.

1.4.2.4 Identification and Authentication

The TSF supports passwords consisting of alphanumeric and all printable ASCII characters, as well as SSH public key authentication. The TSF also allows administrators to set a minimum password length and support passwords with 15 characters or more.

The TSF requires all administrative-users to authenticate before allowing the user to perform any actions other than viewing the warning banner.

1.4.2.5 Security Management

The TOE provides management over TLS, SSH, 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 capability of being updated, and to verify updates via digital signature.

The TSF includes four administrative roles within the Authorized Administrator role: Internal Administrator, Array Administrator, Storage Administrator, and Read-Only Administrator. All roles are considered authorized administrators for the remainder of this document. The device ships with two hard-coded users, but allows for additional users to be authenticated through the use of Active Directory.

1.4.2.6 Protection of the TSF

The TOE uses several protection methods to ensure correct and secure operation: the TOE runs a suite of self-tests during the initial start-up (upon power on) , it provides a means to verify firmware/software updates using a digital signature mechanism prior to installing those updates, the reading of secret and private keys is not allowed, and the TOE provides reliable time stamps for itself.

1.4.2.7 TOE Access

The TOE, for local and remote interactive sessions, terminates sessions after an Authorized Administrator-specified period of session inactivity. The TOE also 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.

1.4.2.8 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 initiates communication via the trusted channel, and also allows remote IT entities to initiate communication.

The TOE permits remote administrators to initiate a trusted path via SSH and HTTPS/TLS. 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. Those notes specific to the TOE are marked "TOE Application Note;" those taken from the ND Protection Profile are marked "PP Application Note."

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 iterations, 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; paragraph 6.4.1.3.2, "Permitted operations on components" 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 are indicated by a number in parenthesis following the requirement number, e.g., FIA_UAU.1.1(1); the iterated requirement titles are similarly indicated, e.g., FIA_UAU.1(1).

Assignments made by the ST author are identified with **bold text.**

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

Refinements that add text use ***bold and italicized text*** to identified 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

This Security Target is conformant to the Common Criteria Version 3.1r3, CC Part 2 extended [2], and CC Part 3 extended [3].

2.2 Conformance to Protection Profiles

This Security Target claims exact compliance to the Protection Profile for Network Devices, Version 1.1, June 8, 2012 [6], including the Security Requirements for Network Devices Errata #3, Version 1.0, November 3, 2014 [7]. This Protection Profile and Errata will be referred to as NDPP or PP for convenience throughout this Security Target.

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 NDPP are carried forward to this ST;
    - No additional threats have been defined in this ST.
- Organizational Security Policies
    - All OSP defined in the NDPP are carried forward to this ST;

o No additional OSPs have been defined in this ST.

- Assumptions
    - All assumptions defined in the NDPP 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 NDPP are carried forward to this ST.
- All SFRs and SARs defined in the NDPP 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 NDPP have been properly instantiated in this Security Target; therefore, this ST shows strict conformance to the NDPP.

3. Security Problem Definition

3.1 Threats

The following table 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.

Table 1: Threats	
Threat	Description
T.ADMIN_ERROR	An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNDETECTED_ACTIONS	Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.
T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T.USER_DATA_REUSE	User data may be inadvertently sent to a destination not intended by the original sender.

3.2 Organizational Security Policies

The following table 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.

Table 2: Organizational Security Policies	
OSP	Description
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.

Table 3: Assumptions	
Assumption	Description
A.NO_GENERAL_PURPOSE	It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available to the TOE, other than those services necessary for the operation, administration and support of the TOE.

Table 3: Assumptions	
Assumption	Description
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

4. Security Objectives

4.1 Security Objectives for the TOE

Table 4: Security Objectives for the TOE	
TOE Objective	Description
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
O.VERIFIABLE_UPDATES	The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source.
O.SYSTEM_MONITORING	The TOE will provide the capability to generate audit data and send those data to an external IT entity.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.TOE_ADMINISTRATION	The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.
O.RESIDUAL_INFORMATION_CLEARING	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.SESSION_LOCK	The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.

4.2 Security Objectives for the Operational Environment

Table 5: Security Objectives for the Operational Environment	
Objective	Description
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available to the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

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 NDPP.

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 NDPP.

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 NDPP are not signified in this section. Operations performed by the ST are denoted according to the formatting conventions in Section 1.5.

Table 6: Security Functional Requirements

#	SFR	Description
1	FAU_GEN.1	Audit Data Generation
2	FAU_GEN.2	User Audit Association
3	FAU_STG_EXT.1	External Audit Trail Storage
4	FCS_CKM.1	Cryptographic Key Generation (Asymmetric Keys)
5	FCS_CKM_EXT.4	Cryptographic Key Zeroization
6	FCS_COP.1(1)	Cryptographic Operation (Data Encryption/Decryption)
7	FCS_COP.1(2)	Cryptographic Operation (Cryptographic Signature)
8	FCS_COP.1(3)	Cryptographic Operation (Cryptographic Hashing)
9	FCS_COP.1(4)	Cryptographic Operation (Keyed-Hash Message Authentication)
10	FCS_TLS_EXT.1	Transport Layer Security
11	FCS_SSH_EXT.1	Secure Shell
12	FCS_HTTPS_EXT.1	HTTP Security
13	FCS_RBG_EXT.1	Extended: Cryptographic Operation: Random Bit Generation
14	FDP_RIP.2	Full Resident Information Protection
15	FIA_PMG_EXT.1	Password Management
16	FIA_UIA_EXT.1	User Identification and Authentication
17	FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanisms
18	FIA_UAU.7	Protected Authentication Feedback
19	FMT_MTD.1	Management of TSF Data (General TSF Data)
20	FMT_SMF.1	Specification of management functions
21	FMT_SMR.2	Security Management Roles
22	FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)
23	FPT_APW_EXT.1	Protection of Administrator Passwords
24	FPT_STM.1	Reliable Time Stamp
25	FPT_TUD_EXT.1	Extended: Trusted Update
26	FPT_TST_EXT.1	Extended: TSF Testing

Table 6: Security Functional Requirements	
# SFR	Description
27 FTA_SSL_EXT.1	TSF-initiated session locking
28 FTA_SSL.3	TSF-initiated termination
29 FTA_SSL.4	User-initiated termination
30 FTA_TAB.1	Default TOE Access Banners
31 FTP_ITC.1	Inter-TSF trusted channel
32 FTP_TRP.1	Trusted Path

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 for 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;
d) Specifically defined auditable events listed in Table 7.

Table 7: Auditable Events			
#	SFR	Auditable Events	Additional Audit Record Contents
1	FAU_GEN.1	None.	
2	FAU_GEN.2	None.	
3	FAU_STG_EXT.1	None.	
4	FCS_CKM.1	None.	
5	FCS_CKM_EXT.4	None.	
6	FCS_COP.1(1)	None.	
7	FCS_COP.1(2)	None.	
8	FCS_COP.1(3)	None.	
9	FCS_COP.1(4)	None.	
10	FCS_TLS_EXT.1	Failure to establish a TSL Session. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
11	FCS_SSH_EXT.1	Failure to establish an SSH session. Establishment/Termination of an SSH session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
12	FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session. Establishment/Termination of a HTTPS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
13	FCS_RBG_EXT.1	None.	
14	FDP_RIP.2	None.	

Table 7: Auditable Events			
#	SFR	Auditable Events	Additional Audit Record Contents
15	FIA_PMG_EXT.1	None.	
16	FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
17	FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
18	FIA_UAU.7	None.	
19	FMT_MTD.1	None.	
20	FMT_SMF.1	None.	
21	FMT_SMR.2	None.	
22	FPT_SKP_EXT.1	None.	
23	FPT_APW_EXT.1	None.	
24	FPT_STM.1	Changes to the time.	The old and new values for the time. Origin of the attempt (e.g., IP address).
25	FPT_TUD_EXT.1	Initiation of update.	No additional information.
26	FPT_TST_EXT.1	None.	
27	FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	No additional information.
28	FTA_SSL.3	The termination of a remote session by the session locking mechanism.	No additional information.
29	FTA_SSL.4	The termination of an interactive session.	No additional information.
30	FTA_TAB.1	None.	
31	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.
32	FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the claimed user identity.

PP Application Note:

The ST author can include other auditable events directly in the table; they are not limited to the list presented.

Many auditable aspects of the SFRs included in this document deal with administrative actions. Item c above requires all administrative actions to be auditable, so no additional specification of the auditability of these actions is specified in Table 7.

Assurance Activity:

The evaluator shall check the administrative guide and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP is described and that the description of the fields contains the information required in FAU_GEN.1.2, and the additional information specified in Table 7.

The evaluator shall also make a determination of the administrative actions that are relevant in the context of the NDPP. The evaluator shall examine the administrative guide and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the PP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are security relevant with respect to the NDPP. The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements.

The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in Table 7 and administrative actions. This should include all instances of an event--for instance, if there are several different I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of the NDPP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.

Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.

FAU_GEN.1.2

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

- Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
- For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, information specified in column three of Table 7.

PP Application Note:

As with the previous component, the ST author should update Table 7 above with any additional information generated. "Subject identity" in the context of this requirement could either be the administrator's user id or the affected network interface, for example.

Assurance Activity:

This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

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.

Assurance Activity:

This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

6.1.1.3 FAU_STG_EXT.1 External Audit Trail 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 implementing the TLS protocol.

PP Application Note:

For applications of the NDPP to TOEs that do not act as audit servers, the TOE relies on a non-TOE audit server for storage and review of audit records. Although the TOE generates audit records, the storage of these audit records and the ability to allow the administrator to review these audit records is provided by the operational environment. The ST author chooses the first clause of the first selection in these cases. The NDPP can also be used to specify requirements for an audit server; in this case, the second clause of the first selection is used.

In the second selection, the ST author chooses the means by which this connection is protected. The ST author also ensures that the supporting protocol requirement matching the selection is included in the ST.

Assurance Activity:

For both types of TOEs (those that act as an audit server and those that send data to an external audit server), there is some amount of local storage. The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access. The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server (for TOEs that are not acting as an audit log server). For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store periodically by sending the data to the audit server.

TOE acts as audit server:

The evaluator shall examine the TSS to ensure it describes the connection supported from non-TOE entities to send the audit data to the TOE, and how the trusted channel is provided. Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel with the TOE, as well as describe any requirements for other IT entities to connect and send audit data to the TOE (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with other IT entities. The evaluator shall perform the following test for this requirement:

- Test 1: The evaluator shall establish a session between an external IT entity and the TOE according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the IT entity and the TOE during several activities of the TOE. The evaluator shall observe that these data are not able to be viewed in the clear during this

transfer, and that they are successfully received by the TOE. The evaluator shall perform this test for each protocol selected in the second selection.

TOE is not an audit server:

The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided. Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server. The evaluator shall perform the following test for this requirement:

- Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation (for asymmetric keys)

FCS_CKM.1.1

The TSF shall generate asymmetric cryptographic keys used for key establishment in accordance with

- NIST Special Publication 800-56A. "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes;
- NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes

and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

PP Application Note:

This component requires that the TOE be able to generate the public/private key pairs that are used for key establishment purposes for the various cryptographic protocols used by the TOE (e.g., IPsec). If multiple schemes are supported, then the ST author should iterate this requirement to capture this capability. The scheme used will be chosen by the ST author from the selection.

Since the domain parameters to be used are specified by the requirements of the protocol in the NDPP, it is not expected that the TOE will generate domain parameters, and therefore there is no additional domain parameter validation needed when the TOE complies to the protocols specified in the NDPP.

SP 800-56B references (but does not mandate) key generation according to FIPS 186-3. For purposes of compliance in this version of the NDPP, RSA key pair generation according to FIPS 186-2 or FIPS 186-3 is allowed in order for the TOE to claim conformance to SP 800-56B.

The generated key strength of 2048-bit DSA and rDSA keys need to be equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.

Assurance Activity:

The evaluator shall use the key pair generation portions of "The FIPS 186-3 Digital Signature Algorithm Validation System (DSA2VS)", "The FIPS 186-3 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and either "The RSA Validation System (RSAVS)" (for FIPS 186-2) or "The 186-3 RSA Validation System (RSA2VS)" (for FIPS 186-3) as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

The evaluator shall ensure that the TSS contains a description of how the TSF complies with 800-56A and/or 800-56B, depending on the selections made. This description shall indicate the sections in 800-56A and/or 800-56B that are implemented by the TSF, and the evaluator shall ensure that key establishment is among those sections that the TSF claims to implement.

Any TOE-specific extensions, processing that is not included in the documents, or alternative implementations allowed by the documents that may impact the security requirements the TOE is to enforce shall be described.

6.1.2.2 FCS_CKM_EXT.4 Cryptographic Key Zeroization

FCS_CKM_EXT.4.1

The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

PP Application Note:

"Cryptographic Critical Security Parameters" are defined in FIPS 140-2 as "security-related information (e.g., secret and private cryptographic keys, and authentication data such as passwords and PINs) whose disclosure or modification can compromise the security of a cryptographic module."

The zeroization indicated above applies to each intermediate storage area for plaintext key/cryptographic critical security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/cryptographic critical security parameter to another location.

Assurance Activity:

The evaluator shall check to ensure the TSS describes each of the secret keys (keys used for symmetric encryption), private keys, and CSPs used to generate key; when they are zeroized (for example, immediately after use, on system shutdown, etc.); and the type of zeroization procedure that is performed (overwrite with zeros, overwrite three times with random pattern, etc.). If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the zeroization procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are zeroized by overwriting once with zeros, while secret keys stored on the internal hard drive are zeroized by overwriting three times with a random pattern that is changed before each write").

6.1.2.3 FCS_COP.1(1) Cryptographic Operation (for data encryption/decryption)

FCS_COP.1.1(1)

The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm AES operating in CBC, GCM and cryptographic key sizes 128-bits and 256-bits that meets the following:

- FIPS PUB 197, “Advanced Encryption Standard (AES)”
- NIST SP 800-38A, NIST SP 800-38D

PP Application Note:

For the first selection, the ST author should choose the mode or modes in which AES operates to support the cryptographic protocols chosen for FTP_ITC and FTP_TRP. If any other modes are used to support requirements in the ST, those should be filled in through the assignment. For the second selection, the ST author should choose the standards that describe the modes specified in the first selection and the assignment.

Assurance Activity:

The evaluator shall use tests appropriate to the modes selected in the above requirement from "The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)", "The XTS-AES Validation System (XTSVS)", The CMAC Validation System (CMACVS)", "The Counter with Cipher Block Chaining-Message Authentication Code (CCM) Validation System (CCMVS)", and "The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS)" (these documents are available from http://csrc.nist.gov/groups/STM/cavp/index.html) as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

6.1.2.4 FCS_COP.1(2) Cryptographic Operations (for cryptographic signature)

FCS_COP.1.1(2)

The TSF shall perform cryptographic signature services in accordance with a

- RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater

that meets the following:

- FIPS PUB 186-2 or FIPS PUB 186-3, “Digital Signature Standard”

PP Application Note:

As the preferred approach for cryptographic signature, elliptic curves will be required in future publications of the NDPP.

PP Application Note:

The ST Author should choose the algorithm implemented to perform digital signatures; if more than one algorithm is available, this requirement (and the corresponding FCS_CKM.1 requirement) should be iterated to specify the functionality. For the algorithm chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm.

For elliptic curve-based schemes, the key size refers to the $\log_2$ of the order of the base point. As the preferred approach for digital signatures, ECDSA will be required in future publications of the NDPP.

Assurance Activity:

The evaluator shall use the signature generation and signature verification portions of "The Digital Signature Algorithm Validation System” (DSA2VS), "The Elliptic Curve Digital Signature Algorithm Validation System” (ECDSA2VS), and "The RSA Validation System” (RSAVS (for 186-2) or RSA2VS (for 186-

3)) as a guide in testing the requirement above. The Validation System used shall comply with the conformance standard identified in the ST (i.e., FIPS PUB 186-2 or FIPS PUB 186-3). This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

6.1.2.5 FCS_COP.1(3) Cryptographic Operation (for cryptographic hashing)

FCS_COP.1.1(3)

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: FIPS Pub 180-3, “Secure Hash Standard.”

PP Application Note:

The selection of the hashing algorithm must correspond to the selection of the message digest size; for example, if SHA-1 is chosen, then the only valid message digest size selection would be 160 bits.

In subsequent publications of the NDPP, it is likely that SHA-1 will no longer be an approved algorithm for cryptographic hashing.

Assurance Activity:

The evaluator shall use "The Secure Hash Algorithm Validation System (SHAVS)" as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

6.1.2.6 FCS_COP.1(4) Cryptographic Operation (for keyed hash message authentication)

FCS_COP.1.1(4)

The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-SHA-1, SHA-256, SHA-384, SHA-512, key size **160, 256, 384, 512 bits**, and message digest sizes 160, 256, 384, 512 bits that meet the following: FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, “Secure Hash Standard.”

PP Application Note:

In future version of the NDPP, SHA-1 may be removed as a valid hash algorithm. Developers are encouraged to transition to the other listed hash algorithms.

Assurance Activity:

The evaluator shall use "The Keyed-Hash Message Authentication Code (HMAC) Validation System (HMACVS)" as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

6.1.2.7 FCS_TLS_EXT.1 TLS

FCS_TLS_EXT.1.1

The TSF shall implement one or more of the following protocols TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246) supporting the following ciphersuites:

Mandatory Ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:

- 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

PP Application Note:

The ST author must make the appropriate selections and assignments to reflect the TLS implementation.

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then "None" should be selected. If administrative steps need to be taken so that the suites negotiated by the implementation are limited to those in this requirement, the appropriate instructions need to be contained in the guidance called for by AGD_OPE.

The Suite B algorithms (RFC 5430) listed above are the preferred algorithms for implementation. The TLS requirement will be changed in the next version of the NDPP to comply with CNSSP 15 and NIST SP 800-131A.

Assurance Activity:

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements). The evaluator shall also perform the following test:

- Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of a HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
- ~~Test 2: The evaluator shall setup a man-in-the-middle tool between the TOE and the TLS Peer and shall perform the following modifications to the traffic:~~
  - ~~[Conditional: TOE is a server] Modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the server denies the client's Finished handshake message.~~

- ~~[Conditional: TOE is a client] Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.~~
- ~~[Conditional: TOE is a client] If a DHE or ECDHE ciphersuite is supported, modify the signature block in the Server's KeyExchange handshake message, and verify that the client rejects the connection after receiving the Server KeyExchange.~~
- ~~[Conditional: TOE is a client] Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.~~[2]

6.1.2.8 FCS_SSH_EXT.1 SSH

FCS_SSH_EXT.1.1

The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and 5656, 6668.

PP Application Note:

The ST author selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted).

In the next version of the NDPP, a requirement will be added regarding rekeying. The requirement will read "The TSF shall ensure that the SSH connection be rekeyed after no more than 2^{28} packets have been transmitted using that key."

FCS_SSH_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.

Assurance Activity:

The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSH_EXT.1.5, and ensure that password-based authentication methods are also allowed. The evaluator shall also perform the following tests:

- Test 1: The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection. Any configuration activities required to support this test shall be performed according to instructions in the operational guidance.
- Test 2: Using the operational guidance, the evaluator shall configure the TOE to accept password-based authentication, and demonstrate that a user can be successfully authenticated to the TOE over SSH using a password as an authenticator.

FCS_SSH_EXT.1.3

[2] This test was stricken per TD #6.

The TSF shall ensure that, as described in RFC 4253, packets greater than **262144** bytes in an SSH transport connection are dropped.

PP Application Note:

RFC 4253 provides for the acceptance of "large packets" with the caveat that packets should be of "reasonable length" or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining "reasonable length" for the TOE.

Assurance Activity:

The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled. The evaluator shall also perform the following test:

- Test 1: The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

FCS_SSH_EXT.1.4

The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, AEAD_AES_GCM_128, AEAD_AES_GCM_256.

PP Application Note:

In the assignment, the ST author can select the AES-GCM algorithms, or "no other algorithms" if AES-GCM is not supported. If AES-GCM is selected, there should be corresponding FCS_COP entries in the ST.

Assurance Activity:

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements). The evaluator shall also perform the following test:

- Test 1: The evaluator shall establish a SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

FCS_SSH_EXT.1.5

The TSF shall ensure that the SSH transport implementation uses SSH_RSA, and no other public key algorithms as its public key algorithm(s).

PP Application Note:

Implementations that select only SSH_RSA will not achieve the 112-bit security strength in the digital signature generation for SSH authentication as is recommended in NIST SP 800-131A. Future versions of this profile will likely disallow the option of selecting only SSH_RSA.

Assurance Activity:

The assurance activity associated with FCS_SSH_EXT.1.4 verifies this requirement.

FCS_SSH_EXT.1.6

The TSF shall ensure that data integrity algorithms used in SSH transport connection is hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512.

PP Application Note:

RFC 6668 specifies the use of the sha2 algorithms in SSH.

Assurance Activity:

The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component. The evaluator shall also check the operational guidance to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the "none" MAC algorithm is not allowed). The evaluator shall also perform the following test:

- Test 1: The evaluator shall establish a SSH connection using each of the integrity algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

FCS_SSH_EXT.1.7

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

Assurance Activity:

The evaluator shall ensure that operational guidance contains configuration information that will allow the security administrator to configure the TOE so that all key exchanges for SSH are performed using DH group 14 and any groups specified from the selection in the ST. If this capability is "hard-coded" into the TOE, the evaluator shall check the TSS to ensure that this is stated in the discussion of the SSH protocol. The evaluator shall also perform the following test:

- Test 1: The evaluator shall attempt to perform a diffie-hellman-group1-sha1 key exchange, and observe that the attempt fails. For each allowed key exchange method, the evaluator shall then attempt to perform a key exchange using that method, and observe that the attempt succeeds.

6.1.2.9 FCS_HTTPS_EXT.1 HTTPS

FCS_HTTPS_EXT.1.1

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

PP Application Note:

The ST author must provide enough detail to determine how the implementation is complying with the standard(s) identified; this can be done either by adding elements to this component, or by additional detail in the TSS.

FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1 .

Assurance Activity:

The evaluator shall check the TSS to ensure that it is clear on how HTTPS uses TLS to establish an administrative session, focusing on any client authentication required by the TLS protocol vs. security administrator authentication which may be done at a different level of the processing stack. Testing for

this activity is done as part of the TLS testing; this may result in additional testing if the TLS tests are done at the TLS protocol level.

6.1.2.10 FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)

FCS_RBG_EXT.1.1

The TSF shall perform all random bit generation (RBG) services in accordance with NIST Special Publication 800-90 using CTR_DRBG (AES) seeded by an entropy source that accumulated entropy from a TSF-hardware-based noise source.

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded with a minimum of 256 bits of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

PP Application Note:

NIST Special Pub 800-90B describes the minimum entropy measurement that will probably be required future versions of FIPS-140. If possible this should be used immediately and will be required in future versions of the NDPP.

For the first selection in FCS_RBG_EXT.1.1, the ST author should select the standard to which the RBG services comply (either 800-90B or 140-2 Annex C).

SP 800-90B contains four different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used (if 800-90B is selected), and include the specific underlying cryptographic primitives used in the requirement or in the TSS. While any of the identified hash functions (SHA-1, SHA-224,SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CTR_DRBG are allowed. While any of the curves defined in 800-90B are allowed for Dual_EC_DRBG, the ST author not only must include the curve chosen, but also the hash algorithm used.

For the second selection in FCS_RBG_EXT.1.1, the ST author indicates whether the sources of entropy are software-based, hardware-based, or both. If there are multiple sources of entropy, the ST will elaborate each entropy sources and whether it is hardware- or software-based. Hardware-based noise sources are preferred.

Note that for FIPS Pub 140-2 Annex C, currently only the method described in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3 is valid. If the key length for the AES implementation used here is different than that used to encrypt the user data, then FCS_COP.1 may have to be adjusted or iterated to reflect the different key length. For the selection in FCS_RBG_EXT.1.2, the ST author selects the minimum number of bits of entropy that is used to seed the RBG.

The ST author also ensures that any underlying functions are included in the baseline requirements for the TOE.

For the selection in FCS_RBG_EXT.1.2, the ST author selects the appropriate number of bits of entropy that corresponds to the greatest security strength of the algorithms included in the ST. Security strength is defined in Tables 2 and 3 of NIST SP 800-57A. For example, if the implementation includes 2048-bit RSA (security strength of 112 bits), AES 128 (security strength 128 bits), and HMAC-512 (security strength 256 bits), then the ST author would select 256 bits.

Assurance Activity:

Documentation shall be produced – and the evaluator shall perform the activities – in accordance with Annex D, Entropy Documentation and Assessment.

The evaluator shall also perform the following tests, depending on the standard to which the RBG conforms.

Implementations Conforming to FIPS 140-2, Annex C

The reference for the tests contained in this section is The Random Number Generator Validation System (RNGVS) [RNGVS]. The evaluator shall conduct the following two tests. Note that the "expected values" are produced by a reference implementation of the algorithm that is known to be correct. Proof of correctness is left to each Scheme.

The evaluator shall perform a Variable Seed Test. The evaluator shall provide a set of 128 (Seed, DT) pairs to the TSF RBG function, each 128 bits. The evaluator shall also provide a key (of the length appropriate to the AES algorithm) that is constant for all 128 (Seed, DT) pairs. The DT value is incremented by 1 for each set. The seed values shall have no repeats within the set. The evaluator ensures that the values returned by the TSF match the expected values.

The evaluator shall perform a Monte Carlo Test. For this test, they supply an initial Seed and DT value to the TSF RBG function; each of these is 128 bits. The evaluator shall also provide a key (of the length appropriate to the AES algorithm) that is constant throughout the test. The evaluator then invokes the TSF RBG 10,000 times, with the DT value being incremented by 1 on each iteration, and the new seed for the subsequent iteration produced as specified in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3. The evaluator ensures that the 10,000th value produced matches the expected value.

Implementations Conforming to NIST Special Publication 800-90

The evaluator shall perform 15 trials for the RBG implementation. If the RBG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RBG functionality.

If the RBG has prediction resistance enabled, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90).

If the RBG does not have prediction resistance, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

- Entropy input: the length of the entropy input value must equal the seed length.
- Nonce: If a nonce is supported (CTR_DRBG with no df does not use a nonce), the nonce bit length is one-half the seed length.
- Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is supported, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.
- Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.

6.1.3 User Data Protection (FDP)

6.1.3.1 FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1

The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to all objects.

Assurance Activity:

"Resources" in the context of this requirement are network packets being sent through (as opposed to "to", as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet. The evaluator shall check to ensure that the TSS describes packet processing to the extent that they can determine that no data will be reused when processing network packets. The evaluator shall ensure that this description at a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.

6.1.4 Identification and Authentication (FIA)

6.1.4.1 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1

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

1. 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, 0x22, 0x27, 0x2B-0x2F, 0x3A-0x3F, 0x5B-0x60, 0x7B-0x7E**;
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;

PP Application Note:

The ST author selects the special characters that are supported by TOE; they may optionally list additional special characters supported using the assignment. "Administrative passwords" refers to passwords used by administrators at the local console or over protocols that support passwords, such as SSH and HTTPS.

Assurance Activity:

The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators on the composition of strong passwords, and that it provides instructions on setting the minimum password length. The evaluator shall also perform the following tests. Note that one or more of these tests can be performed with a single test case.

- Test 1: The evaluator shall compose passwords that either meet the requirements, or fail to meet the requirements, in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, rule characteristics, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.

6.1.4.2 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;
- **ARP**
- **ICMP Echo Response**

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.

PP Application Note:

This requirement applies to users (administrators and external IT entities) of services available from the TOE directly, and not services available by connecting through the TOE. While it should be the case that few or no services are available to external entities prior to identification and authentication, if there are some available (perhaps ICMP echo) these should be listed in the assignment statement; otherwise "no other actions" should be selected.

Authentication can be password-based through the local console or through a protocol that supports passwords (such as SSH), or be certificate based (SSH, TLS).

For communications with external IT entities (e.g., an audit server or NTP server, for instance), such connections must be performed in accordance with FTP_ITC.1, whose protocols perform identification and authentication. This means that such communications (e.g., establishing the IPsec connection to the authentication server) would not have to be specified in the assignment, since establishing the connection "counts" as initiating the identification and authentication process.

Assurance Activity:

The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon". The evaluator shall examine the operational guidance to determine that any necessary preparatory steps (e.g., establishing credential material such as pre-shared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the operational guidance provides clear instructions for successfully

logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the operational guidance provides sufficient instruction on limiting the allowed services.

The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method:

- Test 1: The evaluator shall use the operational guidance to configure the appropriate credential supported for the login method. For that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.
- Test 2: The evaluator shall configure the services allowed (if any) according to the operational guidance, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.
- Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.

6.1.4.3 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1

The TSF shall provide a local password-based authentication mechanism, **Active Directory (via LDAP), SSH public key authentication** to perform administrative user authentication.

Assurance Activity:

Assurance activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1.

6.1.4.4 FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1

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

PP Application Note:

"Obscured feedback" implies the TSF does not produce a visible display of any authentication data entered by a user (such as the echoing of a password), although an obscured indication of progress may be provided (such as an asterisk for each character). It also implies that the TSF does not return any information during the authentication process to the user that may provide any indication of the authentication data.

Assurance Activity:

The evaluator shall perform the following test for each method of local login allowed:

- Test 1: The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.

6.1.5 Security Management (FMT)

6.1.5.1 FMT_MTD.1 Management of TSF Data (for general TSF data)

FMT_MTD.1.1

The TSF shall restrict the ability to manage the TSF data to the Security Administrators.

PP Application Note:

The word "manage" includes but is not limited to create, initialize, view, change default, modify, delete, clear, and append. This requirement is intended to be the "default" requirement for management of TSF data; other iterations of FMT_MTD should place different restrictions or operations available on the specifically-identified TSF data. TSF data includes cryptographic information as well; managing these data would include the association of a cryptographic protocol with an interface, for instance.

Assurance Activity:

The evaluator shall review the operational guidance to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of the NDPP is identified, and that configuration information is provided to ensure that only administrators have access to the functions. The evaluator shall examine the TSS to determine that, for each administrative function identified in the operational guidance, those that are accessible through an interface prior to administrator log-in are identified. For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

6.1.5.2 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1

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

- Ability to administer the TOE locally and remotely;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- No other capabilities

PP Application Note:

The TOE must provide functionality for both local and remote administration, as well as the capability for the administrator to verify that updates received came from a trusted source. They must be capable of performing this action using digital signatures, and optionally a published hash. The ST author chooses whether the published hash verification option is available using the first selection, which must match the corresponding selection in FPT_TUD_EXT.1.3. If the TOE offers the ability for the administrator to configure the services available prior to identification or authentication, or if any of the cryptographic functionality on the TOE can be configured, then the ST author makes the appropriate choice or choices in the second selection, otherwise select "no other capabilities."

Assurance Activity:

The security management functions for FMT_SMF.1 are distributed throughout the PP and are included as part of the requirements in FMT_MTD, FPT_TST_EXT, and any cryptographic management functions specified in the reference standards. Compliance to these requirements satisfies compliance with FMT_SMF.1.

6.1.5.3 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1

The TSF shall maintain the roles:

- Authorized 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

- Authorized Administrator role shall be able to administer the TOE locally;
- Authorized Administrator role shall be able to administer the TOE remotely;

are satisfied.

PP Application Note:

FMT_SMR.2.2 requires that user accounts be associated with only one role. However, note that multiple users may have the same role, and the TOE is not required to restrict roles to a single person.

FMT_SMR.2.3 requires that an authorized administrator be able to administer the TOE through the local console and through a remote mechanism (IPsec, SSH, TLS, TLS/HTTPS). For multiple component TOEs, only the TOE components providing the management control and configuration of the other TOE components require a local administration interface.

Assurance Activity:

The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration. In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of the NDPP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.

6.1.6 Protection of the TSF (FPT)

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

FPT_SKP_EXT.1.1

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

PP Application Note:

The intent of the requirement is that an administrator is unable to read or view the identified keys (stored or ephemeral) through "normal" interfaces. While it is understood that the administrator could directly read memory to view these keys, do so is not a trivial task and may require substantial work on the part of an administrator. Since the administrator is considered a trusted agent, it is assumed they would not endeavour in such an activity.

Assurance Activity:

The evaluator shall examine the TSS to determine that it details how any pre-shared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

6.1.6.2 FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1

The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext passwords.

PP Application Note:

The intent of the requirement is that raw password authentication data are not stored in the clear, and that no user or administrator is able to read the plaintext password through "normal" interfaces. An all-powerful administrator of course could directly read memory to capture a password but is trusted not to do so.

In this version of the PP there are no requirements on the method used to store the passwords in non-plaintext form, but cryptographic methods based on the requirements in FCS_COP are preferred. In future versions of the NDPP, FCS_COP-based cryptographic methods that conform to the Level 2 Credential Storage requirements from NIST SP 800-63 will be required.

Assurance Activity:

The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and the method used to obscure the plaintext password data when stored. The TSS shall also detail passwords are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note.

6.1.6.3 FPT_STM.1 Reliable Time Stamps

FPT_STM.1.1

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

Assurance Activity:

The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time. The TSS provides a description of how the time is maintained and considered reliable in the context of each of the time related functions.

The evaluator examines the operational guidance to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the operational guidance instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication.

- Test 1: The evaluator uses the operational guide to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.
- Test2: [conditional] If the TOE supports the use of an NTP server; the evaluator shall use the operational guidance to configure the NTP client on the TOE, and set up a communication

path with the NTP server. The evaluator will observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the operational guidance.

6.1.6.4 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1

The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2

The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3

The TSF shall provide a means to verify firmware/software updates to the TOE using a published hash prior to installing those updates.

PP Application Note:

The digital signature mechanism referenced in the third element is the one specified in FCS_COP.1(2). The published hash referenced is generated by one of the functions specified in FCS_COP.1(3). The ST author should choose the mechanism implemented by the TOE; it is acceptable to implement both mechanisms.

Assurance Activity:

Updates to the TOE either have a hash associated with them, or are signed by an authorized source. If digital signatures are used, the definition of an authorized source is contained in the TSS, along with a description of how the certificates used by the update verification mechanism are contained on the device. The evaluator ensures this information is contained in the TSS. The evaluator also ensures that the TSS (or the operational guidance) describes how the candidate updates are obtained; the processing associated with verifying the digital signature or calculating the hash of the updates; and the actions that take place for successful (hash or signature was verified) and unsuccessful (hash or signature could not be verified) cases. The evaluator shall perform the following tests:

- Test 1: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. Then, the evaluator performs a subset of other assurance activity tests to demonstrate that the update functions as expected. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update.
- Test 2: The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains or produces an illegitimate update, and attempts to install it on the TOE. The evaluator verifies that the TOE rejects the update.

6.1.6.5 FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1

The TSF shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

Assurance Activity:

The evaluator shall examine the TSS to ensure that it details the self tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

The evaluator shall also ensure that the operational guidance describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.

6.1.7 TOE Access (FTA)

6.1.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions:

- terminate the session

after a Security Administrator-specified time period of inactivity.

Assurance Activity:

The evaluator shall perform the following test:

- Test 1: The evaluator follows the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a local interactive session with the TOE. The evaluator then observes that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator then ensures that re-authentication is needed when trying to unlock the session.

6.1.7.2 FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1

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

Assurance Activity:

The evaluator shall perform the following test:

- Test 1: The evaluator follows the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a remote interactive session with the TOE. The evaluator then observes that the session is terminated after the configured time period.

6.1.7.3 FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1

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

Assurance Activity:

The evaluator shall perform the following test:

- Test 1: The evaluator initiates an interactive local session with the TOE. The evaluator then follows the operational guidance to exit or log off the session and observes that the session has been terminated.
- Test 2: The evaluator initiates an interactive remote session with the TOE. The evaluator then follows the operational guidance to exit or log off the session and observes that the session has been terminated.

6.1.7.4 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1

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

PP Application Note:

This requirement is intended to apply to interactive sessions between a human user and a TOE. IT entities establishing connections or programmatic connections (e.g., remote procedure calls over a network) are not required to be covered by this requirement.

Assurance Activity:

The evaluator shall check the TSS to ensure that it details each method of access (local and remote) available to the administrator (e.g., serial port, SSH, HTTPS). The evaluator shall also perform the following test:

- Test 1: The evaluator follows the operational guidance to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

6.1.8 Trusted Path/Channels (FTP)

6.1.8.1 FTP_ITC.1 Inter-TSF-trusted channel

FTP_ITC.1.1

The TSF shall use SSH, TLS to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, authentication server, **Pure Storage Support Server** 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 **authentication server, audit server**.

PP Application Note:

The intent of the above requirement is to use a cryptographic protocol to protect external communications with authorized IT entities that the TOE interacts with to perform its functions. This is not, however, to be used to specify VPN Gateway functionality; a separate VPN Protection Profile should be used in these instances. Protection (by one of the listed protocols) is required at least for communications with the server that collects the audit information. If it communicates with an authentication server (e.g., RADIUS), then the ST author chooses "authentication server" in FTP_ITC.1.1 and this connection must be protected by one of the listed protocols. If other authorized IT entities (e.g., NTP server) are protected, the ST author makes the appropriate assignments (for those entities) and selections (for the protocols that are used to protect those connections). After the ST author has made the selections, they are to select the detailed requirements in Annex C corresponding to their protocol selection to put in the ST. To summarize, the connection to an external audit collection server is required to be protected by one of the listed protocols. If an external authentication server is supported, then it is required to protect that connection with one of the listed protocols. For any other external server, external communications are not required to be protected, but if protection is claimed, then it must be protected with one of the identified protocols.

While there are no requirements on the party initiating the communication, the ST author lists in the assignment for FTP_ITC.1.3 the services for which the TOE can initiate the communication with the authorized IT entity.

The requirement implies that not only are communications protected when they are initially established, but also on resumption after an outage. It may be the case that some part of the TOE setup involves manually setting up tunnels to protect other communication, and if after an outage the TOE attempts to re-establish the communication automatically with (the necessary) manual intervention, there may be a window created where an attacker might be able to gain critical information or compromise a connection.

Assurance Activity:

The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken. The evaluator shall also perform the following tests:

- Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the operational guidance to ensure that in fact the communication channel can be initiated from the TOE.
- Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.
- Test 4: The evaluators shall, for each protocol associated with each authorized IT entity tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected.

Further assurance activities are associated with the specific protocols.

6.1.8.2 FTP_TRP.1 Trusted Path

FTP_TRP.1.1

The TSF shall use SSH, TLS/HTTPS provide a trusted communication path between itself and 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 detection of modification of the communicated data.

FTP_TRP.1.2

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

FTP_TRP.1.3

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

PP Application Note:

This requirement ensures that authorized remote administrators initiate all communication with the TOE via a trusted path, and that all communications with the TOE by remote administrators is performed over this path. The data passed in this trusted communication channel are encrypted as defined the protocol chosen in the first selection. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Annex C corresponding to their selection are copied to the ST if not already present.

Assurance Activity:

The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method. The evaluator shall also perform the following tests:

- Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test 2: For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote administrative sessions without invoking the trusted path.
- Test 3: The evaluator shall ensure, for each method of remote administration, the channel data are not sent in plaintext.

Further assurance activities are associated with the specific protocols.

6.2 Security Assurance Requirements

This Security Target conformant with the assurance requirements specified in the NDPP. The CC Part 3 conformant security assurance requirements are listed in Table 8. The CC Part 3 extended assurance requirements are listed in Section 6.1 as “Assurance Activity” and Section 6.2.1.

Table 8: Assurance Requirements

Assurance Class	Assurance Component	Assurance Components Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative User guidance
Life-cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_IND.1	Independent testing - conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability analysis

6.2.1 Extended Security Assurance Requirements

These requirements are taken directly from the NDPP and augment or modify the existing SARs taken from CC Part 3.

6.2.1.1 ADV_FSP.1 Basic Functional Specification

There are no specific assurance activities associated with these SARs. The functional specification documentation is provided to support the evaluation activities described in Section 6.1, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because the there is insufficient interface information, then an adequate functional specification has not been provided.

6.2.1.2 AGD_OPE.1 Operational User Guidance

Some of the contents of the operational guidance will be verified by the assurance activities in Section 6.1 and evaluation of the TOE according to the CEM. The following additional information is also required.

The operational guidance shall at a minimum list the processes running (or that could run) on the TOE in its evaluated configuration during its operation that are capable of processing data received on the network interfaces (there are likely more than one of these, and this is not limited to the process that "listens" on the network interface). It is acceptable to list all processes running (or that could run) on the TOE in its evaluated configuration instead of attempting to determine just those that process the network data. For each process listed, the administrative guidance will contain a short (e.g., one- or two-line) description of the process' function, and the privilege with which the service runs. "Privilege" includes the hardware privilege level (e.g., ring 0, ring 1), any software privileges specifically associated with the process, and the privileges associated with the user role the process runs as or under.

The operational guidance shall contain instructions for configuring the 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.

The documentation must describe the process for verifying updates to the TOE, either by checking the hash or by verifying a digital signature. The evaluator shall verify that this process includes the following steps:

1. For hashes, a description of where the hash for a given update can be obtained. For digital signatures, instructions for obtaining the certificate that will be used by the FCS_COP.1(2) mechanism to ensure that a signed update has been received from the certificate owner. This may be supplied with the product initially, or may be obtained by some other means.

2. 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).
3. 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.

The TOE will likely contain security functionality that does not fall in the scope of evaluation under the NDPP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.

6.2.1.3 AGD_PRE.1 Preparative Procedures

As indicated in the introduction above, there are significant expectations with respect to the documentation-especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.

6.2.1.4 ALC_CMC.1 Labeling of the TOE

The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. The evaluator shall ensure that this identifier is sufficient for an acquisition entity to use in procuring the TOE (including the appropriate administrative guidance) as specified in the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

6.2.1.5 ATE_IND.1 Independent Testing - Conformance

The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the CEM and the body of the NDPP's Assurance Activities. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluator must document in the test plan that each applicable testing requirement in the ST is covered.

The test plan identifies the platforms to be tested, and for those 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 of each platform to be tested, and any setup that is 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 the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by the NDPP and used by the cryptographic protocols being evaluated (IPsec, TLS/HTTPS, SSH).

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated 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; a fix installed; and then a successful re-run of the tests, the report would show a "fail" and "pass" result (and the supporting details), and not just the "pass" result.

6.2.1.6 AVA_VAN.1 Vulnerability Assessment

As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in network infrastructure devices and the implemented communication protocols in general, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, a test would be suitable at the assurance level of the NDPP. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.

6.3 Security Requirements Rationale

6.3.1 Security Function Requirement to Security Objective Rationale

The following sections present the rationale that demonstrate that the SFRs meet all security objectives for the TOE.

6.3.1.1 Protected Communications

O.PROTECTED_COMMUNICATIONS

To address the issues concerning transmitting sensitive data to and from the TOE described in Section 3.1, Table 1, row "T.UNAUTHORIZED_ACCESS", compliant TOEs will provide encryption for these communication paths between themselves and the endpoint. These channels are implemented using one (or more) of three standard protocols: IPsec, TLS/HTTPS, and SSH. These protocols are specified by RFCs that offer a variety of implementation choices. Requirements have been imposed on some of these choices (particularly those for cryptographic primitives) to provide interoperability and resistance to cryptographic attack. While compliant TOEs must support all of the choices specified in the ST, they may support additional algorithms and protocols. If such additional mechanisms are not evaluated, guidance must be given to the administrator to make clear the fact that they are not evaluated.

In addition to providing protection from disclosure (and detection of modification) for the communications, each of the protocols described in this document (IPsec, SSH, and TLS/HTTPS) offer two-way authentication of each endpoint in a cryptographically secure manner, meaning that even if there was a malicious attacker between the two endpoints, any attempt to represent themselves to either endpoint of the communications path as the other communicating party would be detected. The requirements on each protocol, in addition to the structure of the protocols themselves, provide protection against replay attacks such as those described in Section 3.1, Table 1, row "T.UNAUTHORIZED_ACCESS", usually by including a unique value in each communication so that replay of that communication can be detected.

(FCS_CKM.1, FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_TLS_EXT.1, FCS_SSH_EXT.1 , FCS_HTTPS_EXT.1 , FCS_RBG_EXT.1, FPT_SKP_EXT.1, FTP_ITC.1, FTP_TRP.1)

6.3.1.2 Verifiable Updates

O.VERIFIABLE_UPDATES

As outlined in Section 3.1, Table 1, row "T.UNAUTHORIZED_UPDATE", failure by the Security Administrator to verify that updates to the system can be trusted may lead to compromise of the entire system. A first step in establishing trust in the update is to publish a hash of the update that can be verified by the System Administrator prior to installing the update. In this way, the Security Administrator can download the update, compute the hash, and compare it to the published hash. While this establishes that the update downloaded is the one associated with the published hash, it does not indicate if the source of the update/hash combination has been compromised or can't be trusted. So, there remains a threat to the system. To establish trust in the source of the updates, the system can provide cryptographic mechanisms and procedures to procure the update, check the update cryptographically through the TOE-provided digital signature mechanism, and install the update on the system. While there is no requirement that this process be completely automated, administrative guidance documentation will detail any procedures that must be performed manually, as well as the manner in which the administrator ensures that the signature on the update is valid.

(FPT_TUD_EXT.1, FCS_COP.1(2), FCS_COP.1(3))

6.3.1.3 System Monitoring

O.SYSTEM_MONITORING

In order to assure that information exists that allows Security Administrators to discover intentional and unintentional issues with the configuration and/or operation of the system as discussed in Section 3.1; Table 1; rows "T.ADMIN_ERROR", "T_UNDETECTED_ACTIONS", and "T.UNAUTHROIZED_ACCESS"; compliant TOEs have the capability of generating audit data targeted at detecting such activity. Auditing of administrative activities provides information that may hasten corrective action should the system be configured incorrectly. Audit of select system events can provide an indication of failure of critical portions of the TOE (e.g., a cryptographic provider process not running) or anomalous activity (e.g., establishment of an administrative session at a suspicious time, repeated failures to establish sessions or authenticate to the system) of a suspicious nature.

In some instances there may be a large amount of audit information produced that could overwhelm the TOE or administrators in charge of reviewing the audit information. The TOE must be capable of sending audit information to an external trusted entity, which mitigates the possibility that the generated audit data will cause some kind of denial of service situation on the TOE. This information must carry reliable timestamps, which will help order the information when sent to the external device.

Loss of communication with the audit server is problematic. While there are several potential mitigations to this threat, the NDPP does not mandate that a specific action takes place; the degree to which this action preserves the audit information and still allows the TOE to meet its functionality responsibilities should drive decisions on the suitability of the TOE in a particular environment.

(FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1, FPT_STM.1)

6.3.1.4 TOE Administration

O.TOE_ADMINISTRATION, O.SESSION_LOCK

In order to provide a trusted means for administrators to interact with the TOE, the TOE provides a password-based logon mechanism. The administrator must have the capability to compose a strong password, and have mechanisms in place so that the password must be changed regularly. To avoid attacks where an attacker might observe a password being typed by an administrator, passwords must be obscured during logon. Session locking or termination must also be implemented to mitigate the risk of an account being used illegitimately. Passwords must be stored in an obscured form, and there must be no interface provided for specifically reading the password or password file such that the passwords are displayed in plain text.

(FIA_UIA_EXT.1, FIA_PMG_EXT.1, FIA_UAU.7, FMT_MTD.1, FMT_SMF.1, FMT_SMR.2, FPT_APW_EXT.1, FTA_SSL_EXT.1, FTA_SSL.3)

O.DISPLAY_BANNER

In order to satisfy the policy requiring users to view and consent to an initial access banner prior to accessing the TOE, the TSF displays an Administrator specified advisory notice and consent warning message prior to the establishment of an administrative user session.

FTA_TAB.1

6.3.1.5 Residual Information Clearing

O.RESIDUAL_INFORMATION_CLEARING

In order to counter the threat that user data is inadvertently included in network traffic not intended by the original sender, the TSF ensures that network packets sent from the TOE do not include data "left over" from the processing of previous network information.

(FDP_RIP.2)

6.3.1.6 TSF Self Test

O.TSF_SELF_TEST

In order to detect some number of failures of underlying security mechanisms used by the TSF, the TSF will perform self-tests. The extent of this self testing is left to the product developer, but a more comprehensive set of self tests should result in a more trustworthy platform on which to develop enterprise architecture.

(FPT_TST_EXT.1)

6.3.2 Security Functional Requirement Dependency Rationale

Table 6: Security Functional Requirements maps the dependencies that exist for each SFR. If the column labeled “Dependency Satisfied” shows a dependency that has not been resolved, the rationale is provided in the following section, why this dependency does not apply for the TOE.

6.3.2.1 Rationale for Unsatisfied Dependencies

The FCS_COP.1(1) dependency on FDP_ITC.1, FDP_ITC.2, or FCS_CKM.1; because the NDPP does not specify an SFR to satisfy this dependency. FCS_RBG_EXT.1 provides the TOE with a method of generating symmetric cryptographic keys for FCS_COP.1(1).

The FCS_COP.1(3) dependency on FDP_ITC.1, FDP_ITC.2, or FCS_CKM.1; because cryptographic hash algorithms do not need cryptographic keys to operate.

The FCS_COP.1(4) dependency on FDP_ITC.1, FDP_ITC.2, or FCS_CKM.1; because the NDPP does not specify an SFR to satisfy this dependency.

6.3.3 Security Assurance Requirements Rationale

This ST contains the assurance requirements from the NDPP. The assurance requirements are listed in the "Component" column of Table 9: SAR Component Dependency Mapping. These assurance requirements are specified in CC Part 3.

6.3.3.1 Security requirement dependency analysis

Table 9: SAR Component Dependency Mapping, maps the dependencies that exist for each SAR to demonstrate all SAR dependencies are satisfied.

Table 9: SAR Component Dependency Mapping		
Component	Dependencies	Satisfied
ADV_FSP.1	None	
AGD_OPE.1	ADV_FSP.1	Yes - ADV_FSP.1
AGD_PRE.1	None	
ASE_CCL.1	ASE_INT.1 ASE_ECD.1 ASE_REQ.1	Yes - ASE_INT.1 Yes - ASE_ECD.1 Yes - ASE_REQ.1
ASE_ECD.1	None	
ASE_INT.1	None	
ASE_OBJ.1	None	
ASE_REQ.1	ASE_ECD.1	Yes - ASE_ECD.1
ASE_TSS.1	ASE_INT.1 ASE_REQ.1 ADV_FSP.1	Yes - ASE_INT.1 Yes - ASE_REQ.1 Yes - ADV_FSP.1
ALC_CMC.1	ALC_CMS.1	Yes – ALC_CMS.1
ALC_CMS.1	None	
ATE_IND.1	ADV_FSP.1 AGD_OPE.1 AGD_PRE.1	Yes – ADV_FSP.1 Yes – AGD_OPE.1 Yes – AGD_PRE.1
AVA_VAN.2	ADV_FSP.1 AGD_OPE.1 AGD_PRE.1	Yes - ADV_FSP.1 Yes – AGD_OPE.1 Yes - AGD_PRE.1

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 Operations
- User Data Protection
- Security Management
- Extended Requirements
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

7.1 Security Audit

7.1.1 Audit 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 7: 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 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 calling application also supplies the subject identity and outcome of the event as text within the audit log message. The format of audit log messages is described in operational guidance [1].

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 auditable event.

The CLI allows restricted access only to custom PureStorage binaries, each of which contain calls to syslog when necessary and captures 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 [1].

FAU_GEN.1, FAU_GEN.2

7.1.2 Audit Storage

The TSF secures audit log transmission using syslog over TLS with syslog-ng and OpenSSL.

The TSF contains a custom database that contains user-specific configurations and log entries. The local audit log server maintains a database table of exactly 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). The TSF deletes the oldest audit log and then records the newest audit log entry. Audit records are protected from unauthorized access by the restrictive GUI and CLI which only allows authorized administrators to edit audit-related settings.

FAU_STG_EXT.1

7.2 Cryptographic Operations

The TSF contains an OpenSSL user space library. The crypto module contains the following CAVP certifications:

- OpenSSL FIPS Object Module v2.0.9
  - AES (Cert #3884)
  - SHA-1, 224, 256, 384, 512 (Cert #3206)
  - HMAC SHA-1, 224, 256, 384, 512 (Cert #2524)
  - CTR_DRBG (AES-256) (Cert #1109)
  - RSA (Cert #1980)

FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1

7.2.1 Cryptographic Key Generation

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”.

FCS_CKM.1

7.2.2 Zeroization

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

Table 10: 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 agreement key	Volatile memory (RAM)
EC DH Private - OpenSSL	EC DH (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, LDAPs	Volatile memory (RAM)
Authentication Pre-Shared Key	Plaintext ASCII string for LDAP authentication	Volatile memory (RAM), Configuration Database

		(Array-Connected SSDs)

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

Table 11: 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 agreement key	Volatile memory (RAM)
EC DH Public - OpenSSL	EC DH (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[3]. 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 keys are zeroized by performing a secure erase. "Secure erase" means using the secure erase command provided by the SSD vendor, as run via a script that directly sends the command to the SSD. There are two possible behaviors that this causes in the drive. First, the drive may physically erase each erase block. Second, the drive may generate a new key to use in writing to the drive, overwriting the old 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.

FCS_CKM_EXT.4

7.2.3 Random Bit Generation

The TOE utilizes the CTR_DRBG (AES 256) mode of SP800-90A for TLS and SSH key exchange. The primitives used include the values of *V* and *Key*, which are described in Section 10.2.1.1 of SP800-90A. The security strength is equal to the AES key size (256). The module provides a 256-bit seed to the DRBG.

Pure Storage's implementation calls RDRAND from a single thread and is guaranteed to cause reseeding every four RDRAND invocations (See Intel® Digital Random Number Generator Software Implementation Guide, Rev. 1.1). The implementation calls RDRAND four times, places the output into a buffer, and performs an AES-CBC-MAC operation on the buffer to returns the high-order bytes (bytes 24-31) as entropy. This produces an expected eight-bits per byte of entropy. It then concatenates four of these individually reseeded 64-bit values in order to produce the 256-bit seed used by the SP 800-90A DRBG. The SP 800-90A DRBG is used to generate, at largest, 256-bit keys that are output by the module.

FCS_RBG_EXT.1

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

7.2.4 TLS

The TSF implements the server and client side of TLSv1.1 and TLSv1.2 according to RFCs 4346 and 5246 respectively. The TSF also implements the extension specified in RFC 5289.

The TSF supports the following TLS cipher suites:

- 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

FCS_TLS_EXT.1

7.2.5 SSH

The TSF uses OpenSSH 6.6p1-2ubuntu2 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 and 6668. It will validate the SSH packet length field and drop the connection after a packet is received whose size exceeds 262,144 bytes. No optional characteristics are supported other than the algorithms listed below.

The following authentication methods and algorithms are supported by the TOE's SSH implementation:

- Password-based
- SSH_RSA (public-key)

The following data encryption algorithms are supported by the TOE's SSH implementation:

- AES-CBC-128
- AES-CBC-256
- AEAD_AES_128_GCM
- AEAD_AES_256_GCM

The following data integrity algorithms are supported by the TOE's SSH implementation:

- hmac-sha1
- hmac-sha1-96
- hmac-sha2-256
- hmac-sha2-512

The following key exchange algorithms are supported by the TOE's SSH implementation:

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

- ecdh-sha2-nistp521

All SSH cryptographic settings are hardcoded into the TOE and not configurable by the administrator.

FCS_SSH_EXT.1

7.2.6 HTTPs

The TSF implements the server side of the HTTPs protocol according to RFC 2818 by using a TLS connection in place of a TCP connection. The TSF listens on port 443 for HTTPs connections through the use of Nginx. The TSF uses HTML over HTTPs to present the administrative users with a secure management interface described in Section 7.2. 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 as described in Section 7.4; no TLS certificate authentication is performed.

FCS_HTTPS_EXT.1

7.3 User Data Protection

The TSF ensures that data will not be reused when processing network packets by overwriting previous buffer contents upon allocation of a new buffer. The TSF accomplishes this by allocating the exact buffer size for each write/addition to each buffer. This ensures that the previous contents are immediately overwritten. The TSF continues to enlarge the buffer by the exact size of each additional write to progressively "grows" the buffer as necessary.

FDP_RIP.2

7.4 Identification and Authentication

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

When a user connects to the console interface, the TSF prompts the user for a username and password. The TSF does not echo any 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 described in Section 7.5.

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 described in Section 7.5.

When a user connects to the HTTPs/TLS interface, the user is initially displayed a username/password authentication form. When a user submits a username/password combination, the TSF attempts to authenticate the client. If the username/password match an authorized administrator's credentials, the user is granted access to graphical user interface described in Section 7.5.

For all three authentication modes, the underlying Linux PAM library is used to authenticate the user in the operational environment. The TOE first checks the local database, and then if the user is not found,

queries the configured Active Directory authentication server. 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 first locally verify that their public key matches, and then submit a request to the Active Directory server to check if their account is allowed to access the TOE.

The TSF can be configured to require passwords of 15 characters or greater. The TSF supports passwords with any printable ASCII character (ASCII codes 32-126).

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

FIA_UIA_EXT.1, FIA_UAU.7

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.

FIA_PMG_EXT.1, FIA_UAU_EXT.2

7.5 Security Management

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

FMT_MTD.1

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 8.1.10.v20130312 and Nginx 1.4.6-1ubuntu3.1. For SSH and the local console, the user is presented with a restrictive CLI that is 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. All other users are authenticated through AD (Active Directory) and are defined by the AD server as an Array Administrator, Storage Administrator, or Read-Only Administrator. The TOE uses the Linux pam_ldap library to authenticate users via LDAPs (LDAP over TLS).

FMT_SMF.1, FMT_SMR.2

7.6 Protection of the TSF

The GUI is restrictive by design and only allows the administrator access to pre-determined functions, and thus ensure that pre-shared keys, symmetric keys, and private keys are inaccessible. The CLI uses an rbash prompt which restricts users to a specific directory and only allows them access to pre-defined binaries created by PureStorage. A SHA-512 hash of the os76 and pureuser account is stored by the TSF for authentication purposes. Access to these hashes and public keys are denied to all accounts except root.

FPT_SKP_EXT.1, FPT_APW_EXT.1

The following TSF security functions utilize the time:

- Audit timestamps
- GUI session timeout
- SSH session timeout
- Console session timeout

The TSF contains a real-time clock to maintain the time between updates from the NTP server. All TSF security functions use the local real-time clock. The TSF uses the NTP daemon to correct the real-time clock, both the NTP daemon and real-time clock are included with the underlying Linux operating system.

FPT_STM.1

Updates to the TOE have a published SHA hash associated with them. Updates are initiated and installed via the internal administrator role. The associated hashes are published in the update release notes. The authorized administrator (root) logs into the TOE via SSH, uploads the update to the TOE via SCP, 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 [1].

FPT_TUD_EXT.1

Upon power-up, the TSF performs a SHA-1 of the kernel, all executables, and all interpreted files. The TSF also performs a known answer test on each cryptographic algorithm. The TSF then begins normal operation, if all of the executables are unchanged and the cryptographic algorithms are operating correctly. These tests demonstrate the correct operation of the device by ensuring that no software modifications have been made, 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. During normal operation the module performs continuous self-tests including: NDRNG continuous self-test, DRBG continuous self-test, and SP800-90A health-tests.

FPT_TST_EXT.1

7.7 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 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, FTA_SSL.4

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.

FTA_SSL_EXT.1, FTA_SSL.3,

7.8 Trusted Path/Channels

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

The TSF uses OpenSSL and the underlying Linux PAM library to communicate with remote authentication servers via TLS.

The TSF uses OpenSSL and OpenSSH to communicate with remote PureStorage Support Servers via SSH.

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

FTP_ITC.1

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

8. Terms and Definitions

Table 12: TOE Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
AD	Active Directory
AES	Advanced Encryption Standard
ARP	Address Resolution Protocol
ASCII	American Standard Code for Information Interchange
CAVP	Cryptographic Algorithm Validation Program (CAVP)
CBC	Cipher Block Chaining
CSP	Critical Security Parameter
CTR	Counter
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
ECDH	Elliptic-Curve Diffie-Hellman
ECDSA	Elliptic-Curve Digital Signature Algorithm
FTP	File Transfer Protocol
GCM	Galois/Counter Mode
GUI	Graphical User Interface
HMAC-SHA	Hashed Message Authentication Code - Secure Hash Algorithm
HTTPs	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IE	Internet Explorer
IP	Internet Protocol
IT	Information Technology
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
NDPP	Network Device Protection Profile
NIST	National Institute of Standards & Technology
RFC	Request For Comment
RS-232	Recommended Standard 232 (computer serial interface, IEEE)
RSA	Rivest, Shamir, & Adleman (public key encryption technology)
SHA	Secure Hash Algorithm
TB	Terabyte
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TTL	Time to Live
UDP	User Datagram Protocol
WAN	Wide Area Network

Table 13: CC Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
CAC	Common Access Card

Table 13: CC Abbreviations and Acronyms	
Abbreviations/ Acronyms	Description
CAP	Composed Assurance Package
CC	Common Criteria
CCRA	Arrangement on the Recognition of Common Criteria Certificates in the field of IT Security
DAC	Discretionary Access Control
DOD	Department of Defense
DOD	See DOD
EAL	Evaluation Assurance Level
IT	Information Technology
OSP	Organizational Security Policy
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SFP	Security Function Policy
SPD	Security Policy Database
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface

9. References

Table 14: TOE Guidance Documentation		
Reference	Description	Date
[1]	PureStorage Guidance Documentation	March 4, 2016

Table 15: Common Criteria v3.1 References			
Reference	Description	Version	Date
[2]	Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model CCMB-2009-07-001	V3.1 R3	July 2009
[3]	Common Criteria for Information Technology Security Evaluation Part 2: Security functional components CCMB-2009-07-002	V3.1 R3	July 2009
[4]	Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components CCMB-2009-07-003	V3.1 R3	July 2009
[5]	Common Criteria for Information Technology Security Evaluation Evaluation Methodology CCMB-2009-07-004	V3.1 R3	July 2009

Table 16: Supporting Documentation			
Reference	Description	Version	Date
[6]	Protection Profile for Network Devices	1.1	June 8, 2012
[7]	Security Requirements for Network Devices Errata #3		November 3, 2014