Security Target

Pulse Secure, LLC Pulse Policy Secure 5.0R13

Document Version 1.10

January 23, 2016

Prepared For:

Pulse Secure, LLC

2700 Zanker Road, Suite 200

San Jose, CA 95134

www.pulsesecure.net

Prepared By:

Apex Assurance Group, LLC

530 Lytton Avenue, Ste. 200

Palo Alto, CA 94301

www.apexassurance.com

Abstract

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), Pulse Secure, LLC Pulse Policy Secure 5.0R13 (formerly known as Junos Pulse Access Control Service 5.0). This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements and the IT security functions provided by the TOE which meet the set of requirements.

Table of Contents

List of Tables

List of Figures

1 Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), Security Target organization, document conventions, and terminology. It also includes an overview of the evaluated product.

1.1 ST Reference

ST Title	Security Target: Pulse Secure, LLC Pulse Policy Secure 5.0R13
ST Revision	1.10
ST Publication Date	January 23, 2016
Author	Apex Assurance Group, LLC

1.2 TOE Reference

TOE Reference	Pulse Secure, LLC Pulse Policy Secure 5.0R13

1.3 Document Organization

This Security Target follows the following format:

SECTION	TITLE	DESCRIPTION
1	Introduction	Provides an overview of the TOE and defines the hardware and software that make up the TOE as well as the physical and logical boundaries of the TOE
2	Conformance Claims	Lists evaluation conformance to Common Criteria versions, Protection Profiles, or Packages where applicable
3	Security Problem Definition	Specifies the threats, assumptions and organizational security policies that affect the TOE
4	Security Objectives	Defines the security objectives for the TOE/operational environment and provides a rationale to demonstrate that the security objectives satisfy the threats
5	Extended Components Definition	Describes extended components of the evaluation (if any)
6	Security Requirements	Contains the functional and assurance requirements for this TOE
7	TOE Summary Specification	Identifies the IT security functions provided by the TOE and also identifies the assurance measures targeted to meet the assurance requirements.

Table 1-1 – ST Organization and Section Descriptions

1.4 Document Conventions

The notation, formatting, and conventions used in this Security Target are consistent with those used in Version 3.1 of the Common Criteria. Selected presentation choices are discussed here to aid the Security Target reader. The Common Criteria allows several operations to be performed on functional requirements: The allowable operations defined in Part 2 of the Common Criteria are *refinement, selection, assignment* and *iteration*.

- Assignment: Indicated with *italicized* text;
- Refinement made by PP author: Indicated with **bold** text and ~~strikethroughs~~, if necessary;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with *italicized and underlined* text;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).

Explicitly stated SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs. When not embedded in a Security Functional Requirement, italicized text is used for both official document titles and text meant to be emphasized more than plain text.

1.5 Document Terminology

The following table describes the acronyms used in this document:

TERM	DEFINITION
AES	Advanced Encryption Standard
CC	Common Criteria version 3.1
CCM	Counter with Cipher Block Chaining-Message Authentication Code
CM	Configuration Management
CSP	Cryptographic security parameter
DES	Data Encryption Standard
DH	Diffie Hellman
FIPS	Federal Information Processing Standard
FIPS-PUB 140-2	Federal Information Processing Standard Publication
FTP	File Transfer Protocol
HMAC	Keyed-Hash Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
IT	Information Technology
NDPP	Network Devices Protection Profile
NIST	National Institute of Standards Technology
OSP	Organizational Security Policy
PP	Protection Profile
RFC	Request for Comment
RNG	Random Number Generator
RSA	Rivest, Shamir, Adelman

TERM	DEFINITION
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function
URL	Uniform Research Locator

Table 1-2 – Acronyms Used in Security Target

1.6 TOE Overview

The TOE is Pulse Secure, LLC Pulse Policy Secure 5.0R13 (formerly known as Junos Pulse Access Control Service 5.0) is a network device that provides a mechanism for authenticating users and assessing the health of their host machines to control network access.

The TOE protects communications between itself and web browsers used for administrator access to TOE management functions using HTTPS/TLS. The TOE includes identification and authentication services to users and supports the use of moderately complex passwords. Users may login locally or remotely. The TOE audits security-relevant events that are associated with activity on the TOE and can store these events on an external syslog server. The TOE protects its own integrity by performing power-on self-tests and the ability to verify the source of updates to the TOE

The Pulse Policy Secure 5.0R13 may also be referred to as the TOE in this document.

1.7 TOE Description

1.7.1 Overview

Pulse Policy Secure 5.0R13 is a software network device TOE. It is delivered as a hardware with bundled software appliance from Pulse Secure. It is also delivered as a virtual appliance.

The TOE provides access controls for 802.1X-enabled switches and access points to provide user access to network, cloud, and other IT resources. It provides session federation for provisioning of remote access user sessions. Role and application policy enforcement provides granular access enforcement.

The TOE stores security-relevant events in local files that can be sent to an external syslog server (via TLS). Auditable events include start-up and shutdown of the audit functions, authentication events, and service requests. Audit records include the date and time, event category, event type, username, and the outcome of the event (success or failure). Local log storage limits are configurable and are monitored. In the event of storage limits being reached the oldest logs will be overwritten.

The TOE includes a FIPS 140-2 validated cryptographic module that provides confidentiality and integrity services for authentication and for protecting communications with adjacent systems. The cryptographic functions are used to support TLS communications with the syslog server and HTTPS/TLS communications with web browsers used for administration.

The TOE protects from residual information retention from network packets sent during administrator sessions by erasing used memory. This ensures that no residual information from packets in a previous information stream can traverse through the TOE. The TOE also clears secret cryptographic keys when they are no longer needed.

The TOE requires administrators to be successfully identified and authenticated using configured user name and password before allowing any administrative access to the TOE functions and data. Passwords can be configured to a minimum of 15 characters and can include special characters. Once authenticated, users are presented with an administrator-defined warning banner and then granted access only to the functions and data define by their user role. Failed login attempts provide only obscured feedback and are logged in the audit logs.

The TOE provides an authorized Administrator role that is responsible for the configuration, maintenance and administrative tasks. The TOE is managed through a web-based administrator console Interface that is accessible both locally and remotely. The administrator can update the TOE and verify the updates using RSA digital signatures provided by the crypto module.

The TOE provides protection mechanisms for cryptographic keys and administrator passwords by storing them in an AES 128 encrypted file system. The TOE provides for both cryptographic and non-cryptographic self-tests in order to ensure the integrity of the TOE software. Also, a reliable system clock is provided by the underlying hardware for use to timestamp audit logs and to determine session timeouts.

The TOE can be configured to terminate interactive user sessions. The administrator can configure the idle timeout interval.

The TOE creates trusted channels between itself and the syslog server using TLS. The TOE uses HTTPS with TLS to establish trusted paths between itself and remote administrators and users that protect the confidentiality and integrity of communications.

1.7.2 Physical Boundary

The TOE is a combined hardware/software TOE and is defined as the Pulse Policy Secure 5.0R13. The TOE boundary is shown below.

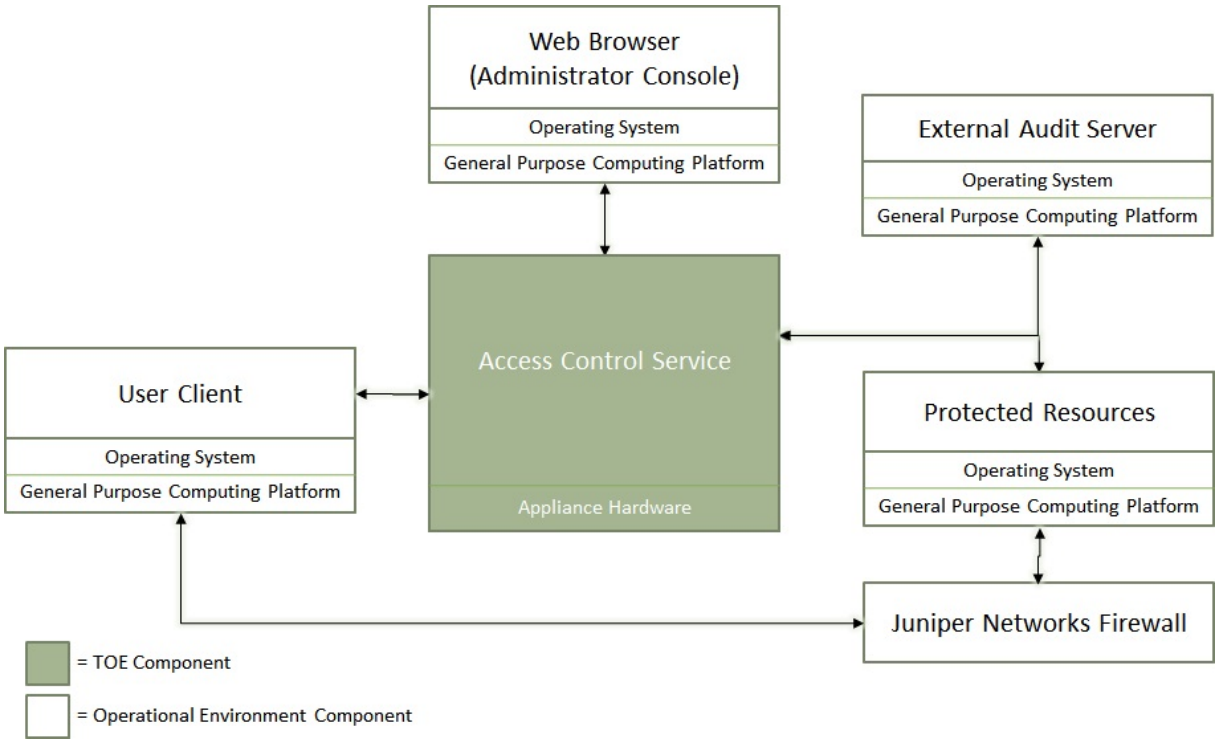

Figure 1 - TOE Boundary

1.7.2.1 *TOE Hardware and Sortware Requirements*

In order to comply with the evaluated configuration, the following hardware and software components shall be used:

PLATFORM	VERSION/MODEL NUMBER
Appliance Hardware Platform	• SM160 (No Cavium) • SM360 • MAG2600 (fixed configuration chassis and blade) • MAG4610 (fixed configuration chassis and blade) Note: SM160 and SM360 are blades that can be run in MAG6610 and MAG6611 chassis

Table 1-3 – Platform Requirements

Virtual Appliance	VERSION/MODEL NUMBER
Virtual Appliance Hardware	IBM BladeCenter 2950 blade server with • 4 CPU Cores, • 4G memory and • 20G disk space
Virtual Appliance Software	VMware vSphere 5.1, 5.0, and 4.1

Table 1-4 – Evaluated Virtual Appliance of the TOE

TOE COMPONENT	VERSION/MODEL NUMBER
Pulse Secure Software	Pulse Policy Secure 5.0R13

Table 1-5 – TOE Component

The TOE interfaces are comprised of the following:

1. Network interfaces which pass traffic
2. Management interface through which handle administrative actions.

1.7.3 Logical Boundary

This section outlines the boundaries of the security functionality of the TOE. The logical boundary of the TOE includes the security functionality described in the following table:

TSF	DESCRIPTION
Security Audit	Auditable events are stored in local files and can be sent to an external syslog server (via TLS). Auditable events include start-up and shutdown of the audit functions, authentication events, and service requests. Audit records include the date and time, event category, event type, username, and the outcome of the event (success or failure). Local log storage limits are configurable and are monitored. In the event of storage limits being reached the oldest logs will be overwritten.
Cryptographic Support	The TOE includes a FIPS 140-2 validated cryptographic module that provides confidentiality and integrity services for authentication and for protecting communications with adjacent systems.
User Data Protection	The TOE is designed to handle commands and information from the remote administrator console. Residual information from these network packets are protected by erasing used memory.
Identification and Authentication	The TOE requires users to provide unique identification and authentication data before any administrative access to the system is granted.
Security Management	The TOE provides an authorized Administrator role that is responsible for the configuration, maintenance and administrative tasks. The TOE is managed through a web-based administrator console Interface that is accessible both locally and remotely.
Protection of the TSF	The TOE provides protection mechanisms for TSF data (e.g. cryptographic keys, administrator passwords). Another protection mechanism is to ensure the integrity of any software/firmware updates are can be verified prior to installation. The TOE provides for both cryptographic and non-cryptographic self-tests, and is capable of automated recovery from failure states. Also, reliable timestamp is made available by the TOE.
TOE Access	The TOE can be configured to terminate interactive user sessions, and to present an access banner with warning messages prior to authentication.

TSF	DESCRIPTION
Trusted Path/Channels	The TOE creates trusted channels between itself and remote trusted authorized IT product (e.g. syslog server) entities that protect the confidentiality and integrity of communications. The TOE creates trusted paths between itself and remote administrators and users that protect the confidentiality and integrity of communications.

Table 1-6 – Logical Boundary Descriptions

1.7.4 Summary of Out-of-Scope Items

The following items are out of the scope of the evaluation:

- None

1.7.5 TOE Security Functional Policies

Since the NDPP does not require it, the TOE does not support any Security Functional Policy.

1.7.6 TOE Product Documentation

The TOE includes the following product documentation:

- *Pulse Policy Secure Administration Guide Version 5.0, April 21, 2015*
- *Operational User Guidance and Preparative Procedures, Pulse Secure, LLC Pulse Policy Secure Version 1.2, September 9, 2015*

2 Conformance Claims

2.1 CC Conformance Claim

The TOE is Common Criteria Version 3.1 Revision 3 (July 2009) Part 2 extended and Part 3 conformant.

2.2 Protection Profile Conformance Claim

The TOE claims conformance to the following U.S. Government approved Protection Profiles (PP):

- Security Requirements for Network Devices, Version 1.1, 08 June 2012 (NDPP)
- Security Requirements for Network Devices Errata #3, 3 November 2014

2.2.1 TOE Type Consistency

Both the PP and the TOE describe network device systems.

2.2.2 Security Problem Definition Consistency

This ST claims exact conformance to the referenced PP. The threats, assumptions, and organizational security policies in the ST are identical to the threats, assumptions, and organizational security policies in the PP.

2.2.3 Security Objectives Consistency

This ST claims exact conformance to the objectives in the referenced PP. No additions or deletions to the objectives have been made. All objectives are consistent with the PP.

2.2.4 Security Functional Requirements Consistency

This ST claims exact conformance to the security functional requirements in the referenced PP.

2.2.5 Security Assurance Requirements Consistency

This ST claims exact conformance to the security assurance requirements in the referenced PP.

2.3 Package Claim

The TOE claims conformance to Security Requirements for Network Devices, Version 1.1, 08 June 2012 and no other assurance or functional packages.

3 Security Problem Definition

The security problem to be addressed by the TOE is described by threats and policies that are common to network devices, as opposed to those that might be targeted at the specific functionality of a specific type of network device, as specified in [NDPP].

This chapter identifies assumptions as A.assumption, threats as T.threat and policies as P.policy.

Note that the assumptions, threats, and policies are the same as those found in [NDPP] such that this TOE serves to address the Security Problem.

3.1 Threats

The following threats are addressed by the TOE, as detailed in table 4 of [NDPP] Annex A.

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.

Table 3-1 – Threats from the NDPP addressed by the TOE

3.2 Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. The TOE is required to meet the following organizational security policies, as specified in table 5 of [NDPP] Annex A:

POLICY	DESCRIPTION

POLICY	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.

Table 3-2 – Organizational Security Policies

3.3 Assumptions

This section contains assumptions regarding the security environment and the intended usage of the TOE, as specified in table 3 of [NDPP] Annex A.

ASSUMPTION	DESCRIPTION
A.NO_GENERAL_PURPOSE	It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
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.

Table 3-3 – Assumptions from the NDPP

4 Security Objectives

4.1 Security Objectives for the TOE

The IT Security Objectives for the TOE are detailed below, as specified in table 6 of [NDPP] Annex A.

OBJECTIVES	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.

Table 4-1 – TOE Security Objectives

4.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are detailed below, as specified in table 7 of [NDPP] Annex A.

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

Table 4-2 – Operational Environment Security Objectives from NDPP.

4.3 Security Objectives Rationale

The objectives for the TOE and operational environment are the same as those specified in [NDPP]. In addition, the statements of threats, assumptions, OSPs and security objectives provided in this security target are the same as those defined in the [NDPP]. Thus the rationales provided in the prose of [NDPP] Section 3 in the tables in [NDPP] Annex A are wholly applicable to this security target.

5 Extended Components Definition

The following extended components are defined by the NDPP. The definition of these components is given in NDPP.

- FAU_STG_EXT.1
- FCS_CKM_EXT.4
- FCS_RBG_EXT.1
- FCS_HTTPS_EXT.1
- FCS_TLS_EXT.1
- FIA_PMG_EXT.1
- FIA_UIA_EXT.1
- FIA_UAU_EXT.5
- FPT_SKP_EXT.1
- FPT_APW_EXT.1
- FPT_TUD_EXT.1
- FPT_TST_EXT.1
- FTA_SSL_EXT.1

5.1 Rationale for Extended Components

This ST includes these extended components to conform to the NDPP requirements.

6 Security Requirements

The security requirements that are levied on the TOE and the Operational environment are specified in this section of the ST.

6.1 Security Functional Requirements

This section specifies the security functional requirements (SFRs) for the TOE, organized by CC class as specified in [NDPP].

The following table identifies all the SFR's implemented by the TOE.

CLASS HEADING	CLASS_FAMILY	DESCRIPTION
Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User Identity Association
	FAU_STG_EXT.1	External Audit Trail Storage
Cryptographic Support	FCS_CKM.1	Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM_EXT.4	Cryptographic Key Zeroization
	FCS_COP.1(1)	Cryptographic Operation (for data encryption/decryption)
	FCS_COP.1(2)	Cryptographic Operation (for cryptographic signature)
	FCS_COP.1(3)	Cryptographic Operation (for cryptographic hashing)
	FCS_COP.1(4)	Cryptographic Operation (for keyed-hash message authentication)
	FCS_RBG_EXT.1	Extended: Cryptographic Operation (Random Bit Generation)
	FCS_HTTPS_EXT.1	Explicit HTTPS
	FCS_TLS_EXT.1	Explicit TLS
User Data Protection	FDP_RIP.2	Full residual information protection
Identification and Authentication	FIA_PMG_EXT.1	User Identification and Authentication
	FIA_UIA_EXT.1	Extended: Password-based Authentication Mechanism
	FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanism
	FIA_UAU.7	Protected Authentication Feedback
Security Management	FMT_MTD.1	Management of TSF Data (for general TSF data)
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.2	Security Roles
Protection of the TSF	FPT_SKP_EXT.1	Extended: Protection of TSF Data (for reading of all symmetric keys)
	FPT_APW_EXT.1	Extended: Protection of Administrator Passwords
	FPT_STM.1	Reliable Time Stamps
	FPT_TUD_EXT.1	Extended: Trusted Update
	FPT_TST_EXT.1	TSF Testing
TOE Access	FTA_SSL_EXT.1	TSF-initiated session locking
	FTA_SSL.3	TSF-initiated termination
	FTA_SSL.4	User-initiated termination

CLASS HEADING	CLASS_FAMILY	DESCRIPTION
	FTA_TAB.1	Default TOE access banners
Trusted Path/Channels	FTP_ITC.1	Inter-TSF trusted channel
	FTP_TRP.1	Trusted path

Table 6-1 – TOE Security Functional Requirements

6.1.1 Security Audit (FAU)

6.1.1.1 *FAU_GEN.1 Audit Data Generation*

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

- Start-up and shut-down of the audit functions;
- All auditable events for the not specified level of audit; and
- All Administrative actions;
- [*Specifically defined auditable events listed in Table 6-2 - Audit Events and Details*].

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 6-2 - Audit Events and Details*].

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL DETAILS
FAU_GEN.1	None.	
FAU_GEN.2	None.	
FAU_STG_EXT.1	None.	
FCS_CKM.1	None.	
FCS_CKM_EXT.4	None.	
FCS_COP.1(1)	None.	
FCS_COP.1(2)	None.	
FCS_COP.1(3)	None.	
FCS_COP.1(4)	None.	
FCS_RBG_EXT.1	None.	

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL DETAILS
FCS_HTTPS_EXT.1	Failure to establish an HTTPS session. Establishment/Termination of an HTTPS session	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_TLS_EXT.1	Failure to establish a TLS session. Establishment/Termination of a TLS session	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FDP_RIP.2	None.	
FIA_PMG_EXT.1	None.	
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the authentication mechanism	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	
FMT_MTD.1	None.	
FMT_SMF.1	None.	
FMT_SMR.2	None.	
FPT_SKP_EXT.1	None.	
FPT_APW_EXT.1	None.	
FPT_STM.1	Changes to the time.	The old and new values for the time. Origin of the attempt (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update.	No additional information.
FPT_TST_EXT.1	Indication that TSF self-test was completed.	Any additional information generated by the tests beyond “success” or “failure”.
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	No additional information
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	No additional information.
FTA_SSL.4	The termination of an interactive session.	No additional information.
FTA_TAB.1	None.	

REQUIREMENT	AUDITABLE EVENTS	ADDITIONAL DETAILS
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.

Table 6-2 - Audit Events and Details from NDPP

6.1.1.2 *FAU_GEN.2 User Identity Association*

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

6.1.1.3 *FAU_STG_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.

6.1.2 Cryptographic Support

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

FCS_CKM.1.1 **Refinement:** 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

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

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.

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

FCS_COP.1.1(1) **Refinement:** 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**

Application Note: AES encryption is used in support of TLS.

6.1.2.4 *FCS_COP.1(2) Cryptographic Operation (for cryptographic signature)*

FCS_COP.1.1(2) **Refinement:** 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, "Digital Signature Standard"**

Application Note: RSA digital signatures are used in support of TLS and software updates (FPT_TUD_EXT.1)

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

FCS_COP.1.1(3) **Refinement:** The TSF shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm **SHA-1** and **message digest sizes 160 bits**. that meet the following: *FIPS Pub 180-3, "Secure Hash Standard."*

Application Note: SHA-1 hashing is used in support of TLS.

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

FCS_COP.1.1(4) **Refinement:** The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm HMAC- **SHA-1**, **key size *160* (in bits) used in HMAC**, **and message digest sizes 160** bits that meet the following: FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."

Application Note: HMAC SHA-1 is used in support of TLS.

6.1.2.7 *FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)*

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with) FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using 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.

6.1.2.8 *FCS_HTTPS_EXT.1 Explicit: HTTPS*

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

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1.

Application Note: HTTPS is used for trusted path communications between the TOE and the remote Administrator Console (FTP_TRP.1).

6.1.2.9 *FCS_TLS_EXT.1 Explicit: TLS*

FCS_ TLS_EXT.1.1 The TSF shall implement one or more of the following protocols TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246] supporting the following cipher suites:

Mandatory Cipher suites:
TLS_RSA_WITH_AES_128_CBC_SHA

Optional Cipher suites:
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA

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.

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

"!", "@", "#", "$", "%", "^", "&", "", "(", ")" and the complete set of standard ASCII characters and control characters;*

2. Minimum password length shall be settable by the Security Administrator, and support passwords of 15 characters or greater;

6.1.4.2 *FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism*

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, *none* to perform user authentication.

6.1.4.3 *User Identification and Authentication (FIA_UIA_EXT.1)*

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

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

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

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.

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.

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.*

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.

6.1.6 Protection of the TSF (FPT)

6.1.6.1 *FPT_SKP_EXT.1Extended: 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.

6.1.6.2 *FPT_APW_EXT.1Extended: 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.

Application Note: The passwords are stored in an AES 128 encrypted file system.

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.

6.1.6.4 *FPT_TUD_EXT.1 Extended: 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 digital signature mechanism prior to installing those updates.

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.

6.1.7 TOE Access

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 interval of session inactivity.

6.1.7.2 *FTA_SSL.3 TSF-initiated Termination*

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

6.1.7.3 *FTA_SSL.4 User--initiated Termination*

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

6.1.7.4 *FTA_TAB.1 Default TOE Access Banners*

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

6.1.8 Trusted Path/Channel (FTP)

6.1.8.1 *FTP_ITC.1Inter-TSF Trusted Channel (Prevention of Disclosure)*

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

FTP_ITC.1.2 The TSF shall permit the TSF, or the **authorized IT entities** to initiate communication via the trusted channel.

FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for *export of audit logs to syslog servers*.

6.1.8.2 *FTP_TRP.1 Trusted Path*

FTP_TRP.1.1	**Refinement:** The TSF shall use **TLS/HTTPS** to 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	**Refinement:** 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.*

6.2 CC Component Hierarchies and Dependencies

This section of the ST demonstrates that the identified SFRs include the appropriate hierarchy and dependencies. This ST follows exactly the security requirements included in the NDPP. Any hierarchies and dependencies are satisfied in accordance with the NDPP.

6.3 Security Assurance Requirements

The Security Assurance Requirements for this evaluation are listed in Section 6.4.3 – Security Assurance Requirements.

6.4 Security Requirements Rationale

6.4.1 Security Functional Requirements

This ST follows exactly the NDPP and all of the security functional requirements within. The NDPP maps SFRs to objectives in Section 3 of the NDPP.

6.4.2 Sufficiency of Security Requirements

The following table presents a mapping of the rationale of TOE Security Requirements to Objectives as described in the NDPP Section 3.

OBJECTIVE	SFR

OBJECTIVE	SFR
Protected Communications O.PROTECTED_COMMUNICATIONS	FCS_CKM.1 FCS_CKM_EXT.4 FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FCS_COP.1(4) FCS_RBG_EXT.1 FCS_HTTPS_EXT.1 FCS_TLS_EXT.1 FTP_ITC.1 FTP_TRP.1
Verifiable Updates O.VERIFIABLE_UPDATES	FPT_TUD_EXT.1 FCS_COP.1(2) FCS_COP.1(3)
System Monitoring O.SYSTEM_MONITORING	FAU_GEN.1 FAU_GEN.2 FAU_STG_EXT.1 FPT_STM.1
TOE Administration O.TOE_ADMINISTRATION O.DISPLAY_BANNER O.SESSION_LOCK	FIA_UIA_EXT.1 FIA_PMG_EXT.1 FIA_UAU_EXT.2 FIA_UAU.7 FMT_MTD.1 FMT_SMF.1 FTA_SSL_EXT.1 FTA_SSL.3 FTA_SSL.4
Residual Information Clearing O.RESIDUAL_INFORMATION_CLEARING	FDP_RIP.2
TSF Self Test O.TSF_SELF_TEST	FPT_TST_EXT.1

Table 6-3 – Rationale for TOE SFRs to Objectives from NDPP

6.4.3 Security Assurance Requirements

The assurance security requirements for this Security Target are from the NDPP. The assurance components are summarized in the following table:

CLASS HEADING	CLASS_FAMILY	DESCRIPTION
ADV: Development	ADV_FSP.1	Basic Functional Specification
AGD: Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative Procedures
ALC: Lifecycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage
ATE: Tests	ATE_IND.1	Independent Testing - Conformance
AVA: Vulnerability Assessment	AVA_VAN.1	Vulnerability Survey

Table 6-4 – Security Assurance Requirements

Detailed assurance activities are described in NDPP Section 4.2.

6.4.4 Security Assurance Requirements Rationale

The ST specifies assurance activities specified in the NDPP.

6.4.5 Security Assurance Requirements Evidence

This section identifies the measures applied to satisfy CC assurance requirements.

SECURITY ASSURANCE REQUIREMENT	EVIDENCE TITLE
ADV_FSP.1 Basic functional specification	Security Target: Pulse Secure, LLC. Pulse Policy Secure 5.0
AGD_OPE.1 Operational User Guidance AGD_PRE.1 Preparative Procedures	Operational User Guidance and Preparative Procedures Supplement: Pulse Secure, LLC. Pulse Policy Secure 5.0
ALC_CMC.1 Labeling of the TOE ALC_CMS.1 TOE CM Coverage	Security Target: Pulse Secure, LLC. Pulse Policy Secure 5.0
ATE_IND.1 Independent Testing	Provided by Evaluation Lab
AVA_VAN.1 Vulnerability Analysis	Provided by Evaluation Lab

Table 6-5 – Security Assurance Rationale and Measures

7 TOE Summary Specification

This section presents the Security Functions implemented by the TOE.

7.1 TOE Security Functions

The security functions performed by the TOE are as follows:

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

7.2 Security Audit

The TOE creates and stores audit records which contain the date and time of the event, type of event, subject (user) identity and the outcome of the event for the following events (the detail of content recorded for each audit event is detailed in Table 6-2 - Audit Events and Details from NDPP.):

- Start-up and shutdown of the audit function;
- Configuration is committed;
- Configuration is changed;
- All use of the identification and authentication mechanisms;
- Service requests;
- Failure to establish an TLS session establishment/termination of an TLS session;
- Changes to the time;
- Initiation of update;
- Indication that TSF self-test was completed;
- Termination of a remote session by the session locking mechanism;
- Termination of an interactive session;
- Initiation/termination/failure of the trusted channel functions.

The TOE provides administrator-defined storage capacity for audit data that are stored locally. The Administrator selects System > Log/Monitoring in the Administrator Console interface, then clicks the Settings tab to display the configuration page. The Max Log Size field accepts the number of megabytes allocated for the log file. When the local audit data store is full, the oldest records are overwritten.

Local audit records are protected against unauthorized access by requiring only authenticated administrators access to the audit data. The Operational Environment protects the physical access to the TOE and the audit data.

Audit data are transferred to the external syslog audit server by the TOE using TLS. The Administrator selects System > Log/Monitoring from the Administrator Console, then clicks the Settings tab to display the configuration page to configure the connection to the syslog server. To set the TLS options, the Administrator selects System > Configuration > Security > SSL Options.

The Security Audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1
- FAU_GEN.2
- FAU_STG_EXT.1

7.3 Algorithms and Certificates

ALGORITHM	Certificate
AES (CBC/GCM)	#2553
rDSA	#1306
SHA	#2153
HMAC	#1573
DH Key Generation	#991
X9.31 RNG	#1212

7.4 Cryptographic Support

The TOE uses a FIPS-validated (CMVP certificate #2012) cryptographic library to provide the TOE will all of its cryptographic functions.

The list of all sections of the appropriate 800-56A standard to which the TOE complies is shown in Section 7.11 NIST SP 800-56 Conformance Statements.

The following table describes each of the secret keys (keys used for symmetric encryption), private keys, and CSPs used to generate key; when they are zeroized.

CSP	Description	How Stored	Where Stored	Zeroization Method
TLS Private Host Key	The first time TLS is configured, the key is generated. Used to identify the host.	Plaintext	Disk	Overwritten three times, first with the byte pattern 0xff, then 0x00, and then 0xff again, before they are deleted
TLS Session Key	Session keys used with TLS, AES 128, 256, HMAC-SHA-1 key (160), DH Private Key	Plaintext	Memory	Scrubbed in memory using OpenSSL scrubbing method overwriting the buffer with random data.
User Password	Plaintext value as entered by user	Hashed	Memory	Overwritten with zero's
RNG State	Internal state and seed key of RNG	Plaintext	Memory	Handled by kernel, overwritten with zero's

Table 7-1 - Key Zeroization Handling

Administrators log in to the TOE to gain access to TOE functions by opening a web browser using TLS/HTTPS configured to use the proper TLS cipher suites (see below). The Administrator then enters the URL for the TOE and identifies and authenticates using valid credentials.

The TOE 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

The TOE does not support any optional TLS characteristics (e.g., extensions supported, client authentication supported).

TLS is also used to secure communications between the TOE and the remote syslog server.

The TOE implements the necessary underlying cryptographic functions in a cryptographic library to support TLS. The TOE generates asymmetric keys in accordance with NIST SP 800-56A. Random bit generation is in accordance with FIPS PUB 140-2 Annex C using a hardware-based noise source. The TOE cryptographic module encrypts and decrypts data using AES in CBC or GCM modes with key sizes 128 or 256 bits. RSA digital signatures with 2048 bits are implemented in accordance with FIPS PUB 186-2. SHA-1 hashing with 160 bit message digests are implemented in accordance with FIPS PUB 180-3. HMAC using SHA-1 is used for keyed hash message authentication in accordance with FIPS 198-1.

The Cryptographic Support function is designed to satisfy the following security functional requirements:

- FCS_CKM.1
- FCS_CKM_EXT.4
- FCS_COP.1(1)
- FCS_COP.1(2)
- FCS_COP.1(3)
- FCS_COP.1(4)
- FCS_RBG_EXT.1
- FCS_HTTPS_EXT.1
- FCS_TLS_EXT.1

7.5 User Data Protection

The only resource made available to information flowing through a TOE is the temporary storage of packet information when remote administrator user access is requested and when information is being routed. Data is not persistent when resources are released by one user/process and allocated to another user/process. Temporary storage (memory) used to build packets is overwritten by the new packet data when the resource is called into use by the next user/process. Therefore, no residual information from packets in a previous information stream can traverse through the TOE.

The User Data Protection function is designed to satisfy the following security functional requirements:

- FDP_RIP.2

7.6 Identification and Authentication

Administrators login to the TOE's Administrator Console by pointing the web browser to the TOE's URL. An HTTPS over TLS connection is established between the web browser and the TOE. The TOE presents a login screen with an administrator-defined access banner where the Administrator enters a user name and password. The TOE verifies the user name and password. Only after verification is the Administrator granted access to the management functions of the TOE by presenting the appropriate web pages on the Administrator Console. The TOE provides only obscured feedback to the administrative user while the authentication is in progress at the local console. Failed login attempts are logged in the audit logs.

The TOE supports administrator password composition to include any combination of upper and lower case letters, numbers, and the following special characters "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", and the complete set of standard ASCII characters and control characters with a minimum length settable by the administrator and support 15 characters or greater.

The Identification and Authentication function is designed to satisfy the following security functional requirements:

- FIA_PMG_EXT.1
- FIA_UIA_EXT.1
- FIA_UAU_EXT.2
- FIA_UAU.7

7.7 Security Management

The TOE only allows authenticated Administrators access to TOE management functions. Based on the Administrator's credentials used at login, the Administrator is granted access to the management functions. TOE functions that are not allowed for the logged in user are not presented in the Administrator Console interface.

Authenticated administrators can administer the TOE locally and remotely, update the TOE, and verify the updates using digital signature capability prior to installing those updates.

The Security Management function is designed to satisfy the following security functional requirements:

- FMT_MTD.1
- FMT_SMF.1
- FMT_SMR.2

7.8 Protection of the TSF

The TOE does not provide an interface to allow any user to view any passwords, pre-shared keys, symmetric keys, and private keys.

The TOE maintains user name and password for authentication purposes. User passwords stored in an AES 128 encrypted file system.

The TOE time function is reliant on the system clock provided by the underlying hardware. The time source is maintained by a reliable hardware clock. The TOE uses system time to timestamp audit log records. It also uses time to determine user session timeouts.

To upgrade the TOE software, the Administrator navigates to Maintenance > System > Upgrade/Downgrade in the Administrator Console interface to show the system software maintenance page. The first step in the installation process (see figure below) is the verification of the RSA (2048 bit) digital signatures are used to verify software updates. The only authorized source of the certificate is Pulse Secure.

The key used for package encryption and decryption is part of the package binary file. The public key, corresponding to the private key that created the signature, is also part of the package binary. The private key used for generating the signature is not part of any package/software that is given to customers. Keys used during the update process are stored in memory.

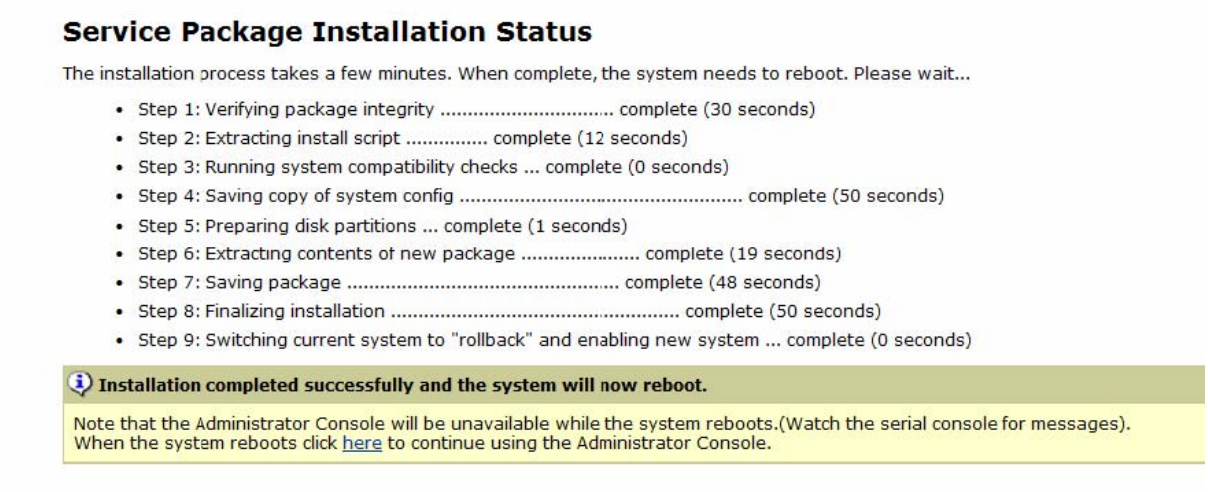

Figure 2 - Software Installation Status

The TOE performs the following hardware self-tests at power-on:

BIOS checks at power-on (words taken from BIOS vendor document)

- Verify boot block checksum. System will hang here if checksum is bad.
- Verify main BIOS checksum.
- Check CMOS diagnostic byte to determine if battery power is OK and CMOS checksum is OK.
- Verify CMOS checksum manually by reading storage area. If the CMOS checksum is bad, update CMOS with power-on default values and clear passwords.

The FIPS 140-2 validated cryptographic module also performs the following known-answer (KAT) power-on self-tests:

ALGORITHM	TEST ATTRIBUTES
Software Integrity	HMAC SHA-1
HMAC	SHA1
AES	Separate encrypt and decrypt
AES CCM	Separate encrypt and decrypt
AES GCM	Separate encrypt and decrypt
RSA	Sign and verify using 2048 bit key
DRBG	CTR_DRBG: AES, 256 bit with and without derivation function HASH_DRBG: SHA256 HMAC_DRBG: SHA256 Dual_EC_DRBG: P-256 and SHA256

ALGORITHM	TEST ATTRIBUTES
X9.31 RNG	128, 256 bit AES keys

To protect the integrity of the TOE software, after the build is done a SHA256 hash of each binary file is created and are put in a file called the manifest file. The contents of the manifest file are in the format of [key, value] pairs. Each line of this file has the key, which is the complete path of a binary file, followed by the corresponding hash. Using the Engineering RSA private key, the manifest file is digitally signed. The manifest.sign file is created containing the SHA256 hash of the whole manifest file and the encrypted hash using the Engineering RSA private key. The manifest and manifest.sign files are included in the package in a directory which will be mounted as the root partition.

At system boot time:

1) The manifest and manifest.sign files are read into memory
2) The signature of manifest file which is in manifest.sign is verified. The verification involves the following steps:
    a. Decrypt the contents of manigest.sign file using engineering RSA public key which is included in the package. The result of this decryption operation is the SHA256 hash of the manifest file at package creation time.
    b. Calculate the SHA256 hash of the whole manifest file in memory
    c. Compare the hashes in (a) and (b) above. If they are the same, then verification is a success. If they are not the same, verification is a failure.
3) If the verification in (2) fails:
    a. A message is logged in the events log at the highest possible critical level indicating that the signature verification of the manifest file failed.
    b. A message is logged in the debug log with details on what failed.
    c. This failure most likely indicates that the file has been tampered with. So there is no point in verifying the hashes of the individual files in the manifest file. So the rest of the file integrity check process is skipped.
4) The individual hashes of the files in the manifest file are verified. This involves the following steps:
    a. Read all the [key, value] pairs from the manifest file.
    b. For each key, which is the path of a binary or shared library, calculate the SHA256 hash of the current file on the system.
    c. Compare the hash in (b) with that stored in the file.
    d. If they are the same then hash verification for this file is a success. If they are not the same, verification for this file is a failure.
    e. If the verification for a file is a failure:
        i. A message is logged in the events log at the highest possible critical level indicating that the hash verification of one of the files failed.
        ii. A message is logged in the debug log with details on what failed, like the exact file that failed.

iii. The failure most likely indicates that the file has been tampered with. So there is no point in verifying the signatures of other files. So the rest of the file integrity verification process is skipped

5) After all the hashes in the manifest file are verified to be correct, a message is logged in the events log indicating that the file integrity check is a success

This process ensures that only authorized software is loaded thus no unauthorized software can be executed ensuring the integrity of the TOE operations.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_SKP_EXT.1
- FPT_APW_EXT.1
- FPT_STM.1
- FPT_TUD_EXT.1
- FPT_TST_EXT.1

7.9 TOE Access

The TOE enables Authorized Administrators to configure an access banner provided with the authentication prompt. The banner can provide warnings against unauthorized access to the TOE as well as any other information that the Authorized Administrator wishes to communicate.

Administrators may access the TOE both locally and remotely only through the Administrator Console using TLS/HTTPS.

User sessions can be locked or terminated by users. The Authorized Administrator can set the TOE so that a user session is terminated after a Administrator-configured period of inactivity.

The TOE Access function is designed to satisfy the following security functional requirements:

- FTA_SSL_EXT.1
- FTA_SSL.3
- FTA_SSL.4
- FTA_TAB.1

7.10 Trusted Path/Channels

The TOE supports and enforces Trusted Channels that protect the communications between the TOE and external syslog audit server from unauthorized disclosure or modification using TLS. It also supports Trusted Paths between itself and remote administrators so that the contents of administrative sessions are protected against unauthorized disclosure or modification using HTTPS with TLS.

The TOE achieves Trusted Paths by use of the TLS protocol which ensures the confidentiality and integrity of user sessions. The encrypted communication path between the TSF and a remote administrator is provided by the use of a TLS session. Remote administrators of the TSF initiate communication with the TSF through the TLS tunnel created by the TLS session. Assured identification is guaranteed by using public key certificate based authentication for TLS. The TLS protocol ensures that the data transmitted over a TLS session cannot be disclosed or altered by using the encryption and integrity mechanisms of the protocol with the FIPS 140-2 validated cryptographic module.

The Trusted Path/Channels function is designed to satisfy the following security functional requirements:

- FTP_ITC.1
- FTP_TRP.1

7.11 NIST SP 800-56 Conformance Statements

The following sections detail all sections of the NIST SP 800-56A standard the TOE complies with for generation of asymmetric cryptographic keys (as claimed in FCS_CKM.1). The relevant sections of 800-56A are section 5.5 “Domain Parameters” and section 5.6 “Private and Public Keys”.

All “SHALL” statements within the listed sections are implemented in the TOE and all “SHALL NOT” statements are adhered to within the TOE and the described functionality/behavior is not present. The implemented option associated with each “SHOULD” and “SHOULD NOT” statement in a referenced section is detailed.

There are no TOE specific extensions relating to cryptographic key generation that are not included in this standard.

7.11.1 Finite Field-Based Key Establishment Schemes

The requirements for Finite Field-Based Key Establishment Schemes are specified in NIST SP 800-56A:

800-56A section	800-56A sub section	Compliance
5.5 Domain Parameters	General	Comply with all “shall” statements.
5.5.1 Domain Parameter Generation	5.5.1.1 FFC Domain Parameter Generation	Comply with all “shall” statements. The FFC parameter is set and so ECC is not used
5.6 Private and Public Keys	General	No statements
5.6.1 Private/Public Key Pair Generation	5.6.1.1 FFC Key Pair Generation	Comply with all “shall” statements. Static and ephemeral public keys used.

800-56A section	800-56A sub section	Compliance
5.6.2 Assurances of the Arithmetic Validity of a Public Key	General	Comply with all "shall" statements. The TOE will determine and explicitly reflect whether or not key establishment is allowed based upon the method(s) of assurance that was used.
	5.6.2.1 Owner Assurances of Static Public Key Validity	Owner Full Validation - The owner performs a successful full public key validation, via pair-wise consistency check
	5.6.2.2 Recipient Assurances of Static Public Key Validity	TTP Generation – The recipient receives assurance that a trusted third party (trusted by the recipient) has generated the public/private key pair in accordance with Section 5.6.1 and has provided the key pair to the owner.
	5.6.2.3 Recipient Assurances of Ephemeral Public Key Validity	Recipient Full Validation - The recipient performs a successful full public key Validation.
	5.6.2.4 FFC Full Public Key Validation Routine	Comply with "shall" statement.
5.6.3 Assurances of the Possession of a Static Private Key	General	Comply with "shall" statement.
	5.6.3.1 Owner Assurances of Possession of a Static Private Key	Owner Receives Assurance via Key Generation - The act of generating a key pair.
5.6.3.2 Recipient Assurance of Owner's Possession of a Static Private Key	General	Comply with all "shall" statements.
	5.6.3.2.1 Recipient Obtains Assurance through a Trusted Third Party	The TOE will be made aware of the method(s) used by the third party.

800-56A section	800-56A sub section	Compliance
	5.6.3.2.2 Recipient Obtains Assurance Directly from the Claimed Owner	The underlying key agreement used by the TOE is "dhOneFlow or (Cofactor) One-Pass Diffie-Hellman". Comply with all "shall" statements.
5.6.4 Key Pair Management	5.6.4.1 Common Requirements on Static and Ephemeral Key Pairs	Comply with all "shall" statements and the "shall not" statement.
	5.6.4.2 Specific Requirements on Static Key Pairs	Comply with all "shall" statements and the "shall not" statement. In item #3 – The TOE will determine whether or not key establishment is allowed based upon the method(s) of assurance that was used.
	5.6.4.3 Specific Requirements on Ephemeral Key Pairs	Comply with all "shall" statements. In item #2 – The TOE will generate an ephemeral key pair just before the ephemeral public key is transmitted. In item #3 – The TOE will determine whether or not to key establishment is allowed based upon the method(s) of assurance that was used.

Table 7-2 **– 800-56A Conformance Statements**