Pulse Policy Secure Security Target

Acumen Security, LLC.

Document Version: 1.1

Prepared For:
Pulse Secure, LLC
2700 Zanker Road, Suite 200
San Jose, CA 95134

Prepared by:
Acumen Security
2400 Research Blvd
Rockville MD 2085

Table Of Contents

1 Security Target Introduction 5
1.1 Security Target and TOE Reference 5
1.2 TOE Overview 5
1.2.1 TOE Product Type 5
1.2.2 TOE Usage 5
1.3 TOE IT Environment 6
1.4 TOE Architecture 7
1.4.1 Physical Boundaries 7
1.4.2 Logical Scope of the TOE 8
1.4.3 Security Functions provided by the TOE 8
1.4.4 TOE Documentation 11
1.4.5 Other References 11
2 Conformance Claims 12
2.1 CC Conformance 12
2.2 Protection Profile Conformance 12
2.3 Conformance Rationale 12
2.3.1 Technical Decisions 12
3 Security Problem Definition 14
3.1 Threats 14
3.2 Assumptions 15
3.3 Organizational Security Policies 16
4 Security Objectives 17
4.1 Security Objectives for the Operational Environment 17
5 Security Requirements 18
5.1 Conventions 19
5.2 Security Functional requirements 19
5.2.1 Security Audit (FAU) 19
5.2.2 Cryptographic Support (FCS) 22
5.2.3 Identification and Authentication (FIA) 26
5.2.4 Security Management (FMT) 28
5.2.5 Protection of the TSF (FPT) 30
5.2.6 TOE Access (FTA) 31
5.2.7 Trusted path/channels (FTP) 31

5.3 TOE SFR Dependencies Rationale for SFRs 32
5.4 Security Assurance Requirements 32
5.5 Rationale for Security Assurance Requirements 33
5.6 Assurance Measures 33
6 TOE Summary Specification 34
7 Terms and Definitions 44

Revision History

Version	Date	Description
1.0	1/20/2020	Initial Release
1.1	3/16/2020	Updated based on ECR

1 Security Target Introduction

1.1 Security Target and TOE Reference

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

Category	Identifier
ST Title	Pulse Policy Secure v9.1 Security Target
ST Version	1.1
ST Date	March 16, 2020
ST Author	Acumen Security, LLC.
TOE Identifier	Pulse Policy Secure v9.1
TOE Hardware	PSA Models 300, 3000, 5000, 7000C, 7000F, and Virtual appliance
TOE Software Version	9.1
TOE Developer	Pulse Secure, LLC 2700 Zanker Road Suite 200 San Jose, CA 95134
Key Words	Network Device, Network Virtual Appliance

Table 1 TOE/ST Identification

1.2 TOE Overview

1.2.1 TOE Product Type

Pulse Policy Secure (PPS) is a next-generation Network Access Control (NAC) that enables visibility to understand an organization's security posture and enforce role-based access and endpoint security policies for network users. PPS allows administrators to define, implement, and enforce policy by enabling endpoint discovery, monitoring, and alerting.

The TOE is classified as a network device (a generic infrastructure device that can be connected to a network) or a virtual network device (a Virtual Appliance that can be connected to a network) depending on the underlying platform. The TOE software consists of Pulse Policy Secure (PPS) 9.1. The appliance's software is built on IVE OS 2.0. The TOE consists of the PPS application, IVE OS, and either the TOE hardware or the VM hypervisor, all of which are delivered with the TOE. The TOE hardware consists of either the PSA Models 300, 3000, 5000, 7000C, or 7000F.

1.2.2 TOE Usage

The TOE is an infrastructure network device that provides secure remote management, auditing, and updating capabilities. The TOE provides secure remote management using an HTTPS/TLS web interface. The TOE generates audit logs and transmits the audit logs to a remote syslog server over a mutually authenticated TLS channel. The TOE verifies the authenticity of software updates by verifying the digital signature prior to installing any update. The TOE software runs as a virtual appliance or on the following hardware: PSA 300, 3000, 5000, 7000C, or 7000F.

The scope of the evaluated functionality includes the following,

- Secure remote administration of the TOE via TLS
- Secure Local administration of the TOE
- Secure connectivity with remote audit servers
- Secure access to the management functionality of the TOE
- Identification and authentication of the administrator of the TOE

No other functionality is included within the scope of this evaluation.

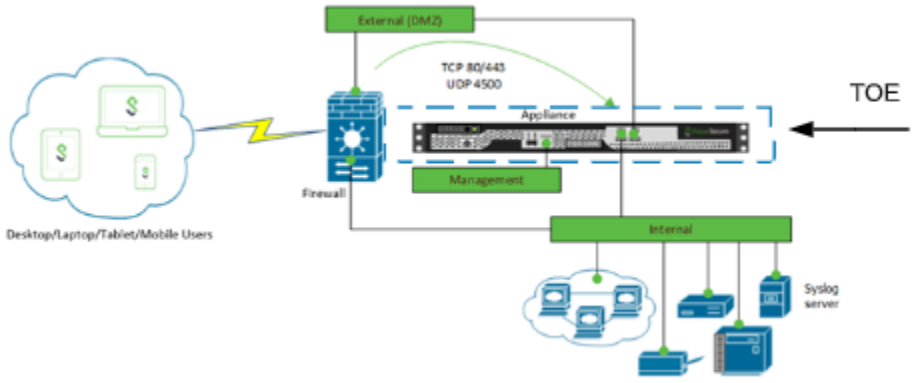

Figure 1 – Pulse Policy Secure Deployment Diagram

In the above diagram, the TOE consists of the appliance within the blue dashed lines. All else is not included within the TOE and is part of the IT environment.

1.3 TOE IT Environment

The TOE's operational environment must provide the following services to support the secure operation of the TOE:

Component	Required	Usage/Purpose Description for TOE performance
Syslog server	Yes	• Conformant with RFC 5424 (Syslog Protocol) • Supporting Syslog over TLS (RFC 5425) • Acting as a TLSv1.1 and/or TLSv1.2 server • Supporting Client Certificate authentication • Supporting at least one of the following cipher suites: ○ TLS_RSA_WITH_AES_128_CBC_SHA ○ TLS_RSA_WITH_AES_256_CBC_SHA ○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA ○ TLS_RSA_WITH_AES_128_CBC_SHA256 ○ TLS_RSA_WITH_AES_256_CBC_ SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ○ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 ○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
Management laptop with web browser	Yes	• Provides remoted management of TOE • • Internet Explorer 11, Google Chrome 50, or Firefox 38 • Supporting TLSv1.1 and/or TLSv1.2 • Supporting Client Certificate authentication • Supporting at least one of the following ciphersuites: ○ TLS_RSA_WITH_AES_128_CBC_SHA

Component	Required	Usage/Purpose Description for TOE performance
		○ TLS_RSA_WITH_AES_256_CBC_SHA ○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA ○ TLS_RSA_WITH_AES_128_CBC_SHA256 ○ TLS_RSA_WITH_AES_256_CBC_ SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ○ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 ○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
CRL Server	Yes	• Conformant with RFC 5280
DNS Sever	Yes	• Conformant with RFC 1035

Table 2 IT Environment Components

1.4 TOE Architecture

1.4.1 Physical Boundaries

The TOE consists of the following hardware:

- PSA 300
- PSA 3000
- PSA 5000
- PSA 7000C
- PSA 7000F

Running:

- Pulse Policy Secure (PPS) v9.1

The PPS software runs on any of the TOE hardware appliance platforms or on a virtual appliance. The TOE is delivered with the PPS v9.1 software installed on one of the PSA appliances. The platforms provide different amounts of processing power and network connectivity options as described in Table 3.

Model	Processor	Network Options
PSA300	Intel® Celeron® J1900	1 x 1 Gigabit Ethernet Eternal Traffic Port 1 x 1 Gigabit Ethernet Internal Traffic/Management Port
PSA3000	Intel® Celeron® J1900	1 x 1 Gigabit Ethernet Eternal Traffic Port 1 x 1 Gigabit Ethernet Internal Traffic/Management Port
PSA5000	Intel® Pentium® G3420	2 x 1 Gigabit Ethernet Eternal Traffic Port 1 x 1 Gigabit Ethernet Internal Traffic/Management Port
PSA7000C	Intel® Xeon® E3-1275V3	2 x 10 Gigabit Ethernet Eternal Traffic Port 1 x 1 Gigabit Ethernet Internal Traffic/Management Port
PSA7000F	Intel® Xeon® E3-1275V3	2 x 10 Gigabit Ethernet Eternal Traffic Port 1 x 1 Gigabit Ethernet Internal Traffic/Management Port

Table 3 TOE Hardware Details

The TOE can also be a virtual appliance on VMware ESXi 6.0, with a Dell PowerEdge R430R530 as the hardware platform. ESXi is a bare-metal hypervisor so there is no underlying operation system. In the evaluated configuration, there are no guest VMs on the physical platform providing non-network device functionality. The virtual appliance platform is described below. The virtual appliance can be download by customers from https://support.pulsesecure.net/ and installed on compliant hardware listed below. License are provided by Pulse Secure via email.

Model	Processor	Hypervisor
Power Edge R430/530	Intel® Xeon® E5-2620 v4	VMware ESXi 6.0

Table 4 VMware Host Details

The guidance documentation that is part of the TOE is listed in Section 1.4.4.

1.4.2 Logical Scope of the TOE

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

1.4.3 Security Functions provided by the TOE

The TOE provides the security functionality required by NDcPP v2.1.

1.4.3.1 Security Audit

The TOE generates audit records for security relevant events. The TOE maintains a local audit log as well as sending the audit records to a remote Syslog server. Audit records sent to the remote server are protected by a TLS connection. Each audit record includes identity (username, IP address, or process), date and time of the event, type of event, and the outcome of the event. The TOE prevents modification to the local audit log.

1.4.3.2 Cryptographic Support

The TOE includes the Pulse Secure Cryptographic Module that 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 TLS and HTTPs connections for secure management and secure connections to a syslog server. TLS and HTTPs are also used to verify firmware updates. The cryptographic services provided by the TOE are described below.

Cryptographic Protocol	Use within the TOE
HTTPS/TLS (client)	Secure connection to syslog FCS_HTTPS_EXT.1, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2
HTPS/TLS (server)	Secure management connections and verification of firmware updates via web browser FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1
AES	Provides encryption/decryption in support of the TLS protocol. FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1
DRBG	Deterministic random bit generation use to generate keys. FCS_TLSS_EXT.1, FCS_RBG_EXT.1

Cryptographic Protocol	Use within the TOE
Secure hash	Used as part of digital signatures and for hashing passwords prior to storage on the TOE. FCS_COP.1/Hash, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1, FPT_APW_EXT.1
HMAC	Provides keyed hashing services in support of TLS. FCS_COP.1/KeyedHash, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1
ECDSA	Provides key generation and signature generation and verification in support of TLS. FCS_CKM.1, FCS_COP.1/SigGen, FCS_COP.1/SigVer, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1
EC-DH	Provides key establishment for TLS. FCS_CKM.2, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1
RSA	Provide key generation and signature generation and verification (PKCS1_V1.5) in support of TLS. FCS_CKM.1, FCS_COP.1/SigGen, FCS_COP.1/SigVer, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1

Table 5 TOE Cryptographic Protocols

Each of these cryptographic algorithms have been validated for conformance to the requirements specified in their respective standards, as identified below.

Algorithm	Standard	Product Name on CAVP Certificate	CAVP Certificate #	Processors
AES 128/256-bit CBC/GCM	ISO 18033-3, ISO 10116 (CBC), IOS 19772 (GCM)	Pulse Secure Cryptographic Module AES, CCM, GCM, XTS	4334	Intel Celeron J1900 Intel PENTIUM G3420 2C/2T Intel Xeon E3-1275v3(x86) Intel Xeon E5 2620 v4
CTR DRBG using AES 256	ISO/IEC 18031:2011	Pulse Secure Cryptographic Module DRBG	1384	Intel Celeron J1900 Intel PENTIUM G3420 2C/2T Intel Xeon E3-1275v3(x86) Intel Xeon E5 2620 v4
ECDSA P-256/P-384	FIPS PUB 286-4, Appendix B.4 (key agreement and digital signature) ISO/IEC 14888-3 (digital signature)	Pulse Secure Cryptographic Module ECDSA	1026	Intel Celeron J1900 Intel PENTIUM G3420 2C/2T Intel Xeon E3-1275v3(x86) Intel Xeon E5 2620 v4
EC-DH	NIST SP 800-56A (key establishment)	Pulse Secure Cryptographic Module ECDH	1270	Intel Celeron J1900 Intel PENTIUM G3420 2C/2T Intel Xeon E3-1275v3(x86) Intel Xeon E5 2620 v4
HMAC-SHA-1/256/384/512	ISO/IEC 9797-2:2011	Pulse Secure Cryptographic Module HMAC	2880	Intel Celeron J1900 Intel PENTIUM G3420 2C/2T Intel Xeon E3-1275v3(x86) Intel Xeon E5 2620 v4
SHA-1/256/384/512	ISO/IEC 10118-3:2004	Pulse Secure Cryptographic Module SHA	3577	Intel Celeron J1900 Intel PENTIUM G3420 2C/2T Intel Xeon E3-1275v3(x86) Intel Xeon E5 2620 v4
RSA 2048/3072	FIPS PUB 186-4 (key generation and Digital Signature) RFC 8017 (key establishment) ISO/IEC 9796-2 (digital signature)	Pulse Secure Cryptographic Module RSA	2345	Intel Celeron J1900 Intel PENTIUM G3420 2C/2T Intel Xeon E3-1275v3(x86) Intel Xeon E5 2620 v4

Table 6 CAVP Algorithm Testing References

1.4.3.3 Identification and Authentication

The TOE authenticates administrative users using a username/password or username/X.509 certificate combination. The TOE does not allow access to any administrative functions prior to successful authentication. The TOE validates and authenticates X.509 certificates for all certificate uses.

The TOE supports passwords consisting of alphanumeric and special characters and enforces minimum password lengths. The TSF supports certificates using RSA or ECDSA signature algorithms.
The TOE only allows users to view the login warning banner, send/receive ICMP packets, and send/receive firewall listening service packets prior to authentication.

Remote administrators are locked out after a configurable number of unsuccessful authentication attempts.

1.4.3.4 Security Management

The TOE allows users with the Security Administrator role to administer the TOE over a remote web UI or a local CLI. These interfaces do not allow the Security Administrator to execute arbitrary commands or executables on the TOE. Security Administrators can manage connections to an external Syslog server, as well as determine the size of local audit storage.

1.4.3.5 Protection of the TSF

The TOE implements several self-protection mechanisms. It does not provide an interface for the reading of secret or private keys. The TOE ensures timestamps, timeouts, and certificate checks are accurate by maintaining a real-time clock. Upon startup, the TOE runs a suite of self-tests to verify that it is operating correctly. The TOE also verifies the integrity and authenticity of firmware updates by verifying a digital signature of the update prior to installing it.

1.4.3.6 TOE Access

The TOE can be configured to display a warning and consent banner when an administrator attempts to establish an interactive session over the local CLI or remote web UI. The TOE also enforces a configurable inactivity timeout for remote and local administrative sessions.

1.4.3.7 Trusted Path/Channels

The TOE uses TLS to provide a trusted communication channel between itself and remote Syslog servers. The trusted channels utilize X.509 certificates to perform mutual authentication. The TOE initiates the TLS trusted channel with the remote server.

The TOE uses HTTPS/TLS to provide a trusted path between itself and remote administrative users. The TOE does not implement any additional methods of remote administration. The remote administrative users are responsible for initiating the trusted path when they wish to communicate with the TOE.

1.4.3.8 Unevaluated Functionality

The TOE includes the following functionality that is not covered in this Security Target and the associated evaluation:

- Network Security and Application Access Control Integration
- Federation
- Guest Access
- Anti-Malware Protection and Patch Assessment
- Firewall Listening Service

These features may be used in the evaluated configuration; however, no assurance as to the correct operation of these features is provided.

1.4.3.9 Excluded Functionality

The TOE includes the following functionality that may not be enabled or used in in the CC evaluated configuration:

- DMI Agent
- SNMP Traps
- External Authentication Servers for administrator authentication

1.4.4 TOE Documentation

The table below lists the TOE guidance documentation.

Reference	Title	Delivery
[CC1]	Pulse Secure Operational User Guidance and Preparative Procedures	Will be available on NIAP website
[AG]	Pulse Policy Secure Administration Guide	Online: PPS Admin Guide
[HW1]	Pulse Policy Secure Supported Platforms Guide	Online: HW Guide
[ST]	Pulse Policy Secure Security Target	Will be available on NIAP website

Table 7 TOE Guidance Documents

1.4.5 Other References

- collaborative Protection Profile for Network Devices, Version 2.1 [NDcPP]

2 Conformance Claims

2.1 CC Conformance

This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 3 extended

2.2 Protection Profile Conformance

This TOE is conformant to:

- collaborative Protection Profile for Network Devices, Version 2.1 [NDcPP]

2.3 Conformance Rationale

This Security Target provides exact conformance to the collaborative Protection Profile for Network Devices, Version 2.1 [NDcPP]. The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profile performing only operations defined there.

2.3.1 Technical Decisions

All NIAP Technical Decisions (TDs) issued to date that are applicable to [NDcPP] have been addressed. The following table identifies all applicable TD:

Identifier	Applicable	Exclusion Rationale (if applicable)
TD0484 – NIT Technical Decision for Interactive sessions in FTA_SSL_EXT.1 & FTA_SSL.3	Yes	
TD0483 – NIT Technical Decision for Applicability of FPT_APW_EXT.1	Yes	
TD0482 – NIT Technical Decision for Identification of usage of cryptographic schemes	Yes	
TD0481 – NIT Technical Decision for FCS_(D)TLSC_EXT.X.2 IP addresses in reference identifiers	Yes	
TD0480 – NIT Technical Decision for Granularity of audit events	Yes	
TD0478 – NIT Technical Decision for Application Notes for FIA_X509_EXT.1 iterations	Yes	
TD0477 – NIT Technical Decision for Clarifying FPT_TUD_EXT.1 Trusted Update	Yes	
TD0475 – NIT Technical Decision for Separate traffic consideration for SSH rekey --	No	FCS_SSHC_EXT.1 is not claimed.
TD0453: NIT Technical Decision for Clarify authentication methods SSH clients can use to authenticate SSH se	No	FCS_SSHC_EXT.1 is not claimed.
TD0451 – NIT Technical Decision for ITT Comm UUID Reference Identifier	Yes	
TD0450 – NIT Technical Decision for RSA-based ciphers and the Server Key Exchange message	Yes	

Identifier	Applicable	Exclusion Rationale (if applicable)
TD0447 – NIT Technical Decision for Using 'diffie-hellman-group-exchange-sha256' in FCS_SSHC/S_EXT.1.7	No	FCS_SSHS_EXT is not claimed.
TD0425 – NIT Technical Decision for Cut-and-paste Error for Guidance AA	Yes	
TD0424 – NIT Technical Decision for NDcPP v2.1 Clarification – FCS_SSHC/S_EXT.1.5	No	SSH is not included in the evaluation.
TD0423 – NIT Technical Decision for Clarification about application of RFI#201726rev2	Yes	
TD0412 – NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy	No	FCS_SSHS_EXT is not claimed.
TD0411 – NIT Technical Decision for FCS_SSHC_EXT.1.5, Test 1 – Server and client side seem to be confused	No	FCS_SSHC_EXT is not claimed.
TD0410 – NIT Technical Decision for Redundant assurance activities associated with FAU_GEN.1	Yes	
TD0409 – NIT Technical Decision for Applicability of FIA_AFL.1 to key-based SSH authentication	No	SSH not claimed.
TD0408 – NIT Technical Decision for local vs. remote administrator accounts	Yes	
TD0407 – NIT Technical Decision for handling Certification of Cloud Deployments	No	TOE is not cloud-based.
TD0402 – NIT Technical Decision for RSA-based FCS_CKM.2 Selection	Yes	
TD0401 – NIT Technical Decision for Reliance on external severs to meet SFRs	Yes	
TD0400 – NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment	Yes	
TD0399 – NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2)	Yes	
TD0398 – NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR	No	FCS_SSHS_EXT and FCS_SSHC_EXT are not claimed.
TD0397 – NIT Technical Decision for Fixing AES-CTR Mode Tests	No	AES-CTR mode is not used in the TOE.
TD0396 – NIT Technical Decision for FCS_TLSC_EXT.1.1, Test 2	Yes	
TD0395 – NIT Technical Decision for Different Handling of TLS 1.1 and TLS 1.2	Yes	

Table 8 Technical Decisions

3 Security Problem Definition

The security problem definition has been taken from [NDcPP] and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any organizational security policies that the TOE is expected to enforce.

3.1 Threats

The following threats are drawn directly from the [NDcPP].

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

T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the network device without administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or device credentials for use by the attacker.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

Table 9 Threats

3.2 Assumptions

The following assumptions are drawn directly from the [NDcPP].

ID	Assumption
A.PHYSICAL_PROTECTION	The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.
A.LIMITED_FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide computing platform for general purpose applications (unrelated to networking functionality).
A.NO_THRU_TRAFFIC_PROTECTION	A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of network devices (e.g, firewall).

A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious administrator that actively works to bypass or compromise the security of the device. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trusted source (aka 'root store', 'trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
A.REGULAR_UPDATES	The network device firmware and software is assumed to be updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The administrator's credentials (private key) used to access the network device are protected by the platform on which they reside.
A.COMPONENTS_RUNNING (applies to distributed TOEs only)	For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

Table 10 Assumptions

3.3 Organizational Security Policies

The following Organizational Security Policies are drawn directly from the [NDcPP].

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

Table 11 OSPs

4 Security Objectives

The security objectives have been taken from [NDcPP] and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the Operational Environment

The following security objectives for the operational environment assist the TOE in correctly providing its security functionality. These track with the assumptions about the environment.

ID	Objective for the Operation Environment
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	TOE Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.COMPONENTS_RUNNING (applies to distributed TOEs only)	For distributed TOEs the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Security Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

Table 12 Objectives for the Operational Environment

5 Security Requirements

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

Requirement	Description
FAU_GEN.1	Audit data generation
FAU_GEN.2	User identity association
FAU_STG.1	Protected audit event storage
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_HTTPS_EXT.1	HTTPS Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_TLSC_EXT.1	TLS Client Protocol
FCS_TLSC_EXT.2	TLS Client Protocol with authentication
FCS_TLSS_EXT.1	TLS Server Protocol
FIA_AFL.1	Authentication Failure Management
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1/Rev	X.509 Certificate Validation
FIA_X509_EXT.2	Certificate Authentication
FIA_X509_EXT.3	Certificate Requests
FMT_MOF.1/Functions	Management of security functions behavior
FMT_MOF.1/ManualUpdate	Management of security functions behaviour
FMT_MTD.1/CoreData	Management of TSF Data
FMT_MTD.1/CryptoKeys	Management of TSF data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on security roles
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_TST_EXT.1	TSF Testing
FPT_TUD_EXT.1	Trusted Update
FPT_STM_EXT.1	Reliable Time Stamps
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination
FTA_TAB.1	Default TOE Access Banner

FTP_ITC.1	Inter-TSF trusted channel
FTP_TRP.1/Admin	Trusted Path

Table 13 SFRs

5.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the conventions within NDcPP and the below conventions within selections or assignments to identify the operations defined by the CC:

- Assignment: Indicated with [*italicized*] text;
- Refinement: Indicated with **bold** text; All bold text from the PP was retained;
- Selection: Indicated with [underlined] text;
- Iteration: Indicated by appending the iteration identifier after a slash, e.g., /SigGen.
- Where operations were completed in the PP itself, the formatting used in the PP has been retained.
- Each selection and assignment from the PP is offset by [brackets], to include selections or assignments within a selection.

Extended SFRs are identified by having a label 'EXT' after the requirement name. Formatting conventions outside of operations matches the formatting specified within the PP.

5.2 Security Functional requirements

5.2.1 Security Audit (FAU)

5.2.1.1 FAU_GEN.1 Audit data generation

FAU_GEN.1.1

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) *All administrative actions comprising:*

- *Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).*
- *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
- *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
- *Resetting passwords (name of related user account shall be logged).*
- [[*Starting and stopping services*], no other actions]*;*

d) *Specifically defined auditable events listed in Table 14*.

FAU_GEN.1.2

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of Table 14*.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG.1	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_RBG_EXT.1	None.	None.
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSC_EXT.2	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate Any addition, replacement or removal of trust anchors in the TOE's trust store.	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Functions	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	Management of cryptographic keys.	None.
FMT_SMF.1	All management activities of TSF data.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1 (if "terminate the session" is selected)	Any attempts at unlocking of an interactive session.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.

Table 14 Security Functional Requirements and Auditable Events

5.2.1.2 FAU_GEN.2 User identity association

FAU_GEN.2.1

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

5.2.1.3 FAU_STG.1 Protected audit trail storage

FAU_STG.1.1

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

FAU_STG.1.2

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

5.2.1.4 FAU_ STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1.1

The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2

The TSF shall be able to store generated audit data on the TOE itself.

- [TOE shall consist of a single standalone component that stores audit data locally].

FAU_STG_EXT.1.3

The TSF shall [overwrite previous audit records according to the following rule: [*the oldest log file is overwritten by the new audit file*]] when the local storage space for audit data is full.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1

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

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;
- ECC schemes using "NIST curves" [P-256, P-384] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;

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

5.2.2.2 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1

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

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

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

ST Application Note
FCS_CKM.2 was updated based on TD0402.

5.2.2.3 FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- *For plaintext keys in volatile storage, the destruction shall be executed by a* [single overwrite consisting of [zeroes]]*;*
- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that:* [
    - logically addresses the storage location of the key and performs a [single, [*three*]-pass] overwrite consisting of a pseudo-random pattern using the TSF's RBG, a new value of the key;
    - instructs a part of the TSF to destroy the abstraction that represents the key]

that meets the following: *No Standard*.

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

FCS_COP.1.1/DataEncryption

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

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

FCS_COP.1.1/SigGen

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

- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [*2048 bits, 3072 bits*],
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [*256 bits, 384 bits*]

]

that meet the following: [

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

].

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

FCS_COP.1.1/Hash

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

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

FCS_COP.1.1/KeyedHash

The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm *[HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384]* and cryptographic key sizes [*128 bits, 160 bits, 256 bits, 384 bits, used in HMAC*] **and message digest sizes [160, 256, 384] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

5.2.2.8 FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1.1

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

FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3

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

5.2.2.9 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1

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

FCS_RBG_EXT.1.2

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

5.2.2.10 FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1

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

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289.]

FCS_TLSC_EXT.1.2

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

ST Application Note
FCS_TLSC_EXT.1.2 was updated based on TD0481.

FCS_TLSC_EXT.1.3

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

- Not implement any administrator override mechanism].

FCS_TLSC_EXT.1.4

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

5.2.2.11 FCS_TLSC_EXT.2 TLS Client Protocol with authentication

FCS_TLSC_EXT.2.1

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

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289.]

FCS_TLSC_EXT.2.2

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

ST Application Note
FCS_TLSC_EXT.2.2 was updated based on TD0481.

FCS_TLSC_EXT.2.3

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

- Not implement any administrator override mechanism].

FCS_TLSC_EXT.2.4

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

FCS_TLSC_EXT.2.5

The TSF shall support mutual authentication using X.509v3 certificates.

5.2.2.12 FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1

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

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289.]

FCS_TLSS_EXT.1.2

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

FCS_TLSS_EXT.1.3

The TSF shall [perform RSA key establishment with key size [2048 bits, 3072 bits]; generate EC Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1], and no other curves].

5.2.3 Identification and Authentication (FIA)

5.2.3.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1

The TSF shall detect when [an Administrator configurable positive integer within [*2 to 2147483647*]] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed].

ST Application Note
FIA_AFL.1 was updated based on TD0408.

5.2.3.2 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [*"!", "@", "#", "$", "%", "^", "&", "*", "(", ")",*[*and standard printable ASCII characters (values 0x20 – 0x7E)*]];
b) Minimum password length shall be configurable to between [*15*] and [*15*] *characters*.

5.2.3.3 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1

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

- Display the warning banner in accordance with FTA_TAB.1;
- [[*Respond to ICMP Echo messages with an ICMP Echo Reply message*.]].

FIA_UIA_EXT.1.2

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

5.2.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1

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

ST Application Note
FIA_UAU_EXT.2 was updated based on TD0408.

5.2.3.5 FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1

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

5.2.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

FIA_X509_EXT.1.1/Rev

The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation **supporting a minimum path length of three certificates**.
- The certificate path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3].
- The TSF shall validate the extendedKeyUsage field according to the following rules:

- *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
- *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
- *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
- *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

FIA_X509_EXT.1.2/Rev

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

5.2.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1

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

FIA_X509_EXT.2.2

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

5.2.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1

The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [Common Name, Organization, Organizational Unit, Country].

FIA_X509_EXT.3.2

The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.2.4 Security Management (FMT)

5.2.4.1 FMT_MOF.1/Functions Management of security functions behavior

FMT_MOF.1/Functions

The TSF shall restrict the ability to [determine the behavior of, modify the behavior of] the functions [transmission of audit data to an external IT entity, handling of audit data] to *Security Administrators.*

5.2.4.2 FMT_MOF.1/ManualUpdate Management of security functions behaviour

FMT_MOF.1.1/ManualUpdate

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

5.2.4.3 FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1.1/CoreData

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

5.2.4.4 FMT_MTD.1/CryptoKeys Management of TSF Data

FMT_MTD.1.1/CryptoKeys

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

5.2.4.5 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 configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- [
    - Ability to configure audit behaviour;
    - Ability to manage cryptographic keys;
    - Ability to configure the cryptographic functionality;
    - Ability to set the time which is used for time-stamps;
    - Ability to import X.509v3 certificates to the TOE's trust store.

]

5.2.4.6 FMT_SMR.2 Restrictions on security roles

FMT_SMR.2.1

The TSF shall maintain the roles:

- *Security Administrator.*

FMT_SMR.2.2

The TSF shall be able to associate users with roles.

FMT_SMR.2.3

The TSF shall ensure that the conditions

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

are satisfied.

5.2.5 Protection of the TSF (FPT)

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

FPT_SKP_EXT.1.1

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

5.2.5.2 FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1

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

FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext administrative passwords.

5.2.5.3 FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1

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

- *BIOS checks*
  - *Verify boot block checksum*
  - *Verify main BIOS checksum*
  - *Check COMS diagnostic byte to determine if battery power is OK and CMOS checksum is OK.*
  - *Verify CMOS checksum manually by reading storage area*
- *Cryptographic library functionality test*
  - *HMAC-SHA-256 integrity check of the library*
  - *HMAC-SHA-1 KAT*
  - *HMAC-SHA-256 KAT*
  - *HMAC-SHA-384 KAT*
  - *AES 128 ECB Encrypt and Decrypt KAT*
  - *AES 256 GCM Encrypt and Decrypt KAT*
  - *RSA 2048 SHA-256 Sign and Verify KAT*
  - *ECDSA P-224 SHA-512 Sign and Verify PCT*
  - *DRBG AES-CTR-256 KAT (invoking the instantiate, reseed, and generate functions)*
- *Firmware integrity checks*
  - *RSA 2048 SHA-512 digital signature verification of the manifest file. This file contains a list of all executables that are part of the TSF*
  - *SHA-256 integrity check of each executable file in the TSF using the pre-calculated hashes from the manifest file*.]

5.2.5.4 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

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

5.2.5.5 FPT_STM_EXT.1 Reliable Time Stamps

FPT_STM_EXT.1.1

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

FPT_STM_EXT.1.2

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

5.2.6 TOE Access (FTA)

5.2.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions, [

- terminate the session]

after a Security Administrator-specified time period of inactivity.

5.2.6.2 FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1

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

5.2.6.3 FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1

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

5.2.6.4 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1

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

5.2.7 Trusted path/channels (FTP)

5.2.7.1 FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1

The TSF shall **be capable of using [TLS] to** provide a trusted communication channel between itself and **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 [*audit server communications*].

5.2.7.2 FTP_TRP.1/Admin Trusted Path

FTP_TRP.1.1/Admin

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

FTP_TRP.1.2/Admin

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

FTP_TRP.1.3/Admin

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

5.3 TOE SFR Dependencies Rationale for SFRs

[NDcPP] contains all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the PP has been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from [NDcPP] which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the table below.

Assurance Class	Components	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

Table 15 Security Assurance Requirements

5.5 Rationale for Security Assurance Requirements

The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.

5.6 Assurance Measures

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by Pulse Secure to satisfy the assurance requirements. The table below lists the details.

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ALC_CMS.1	
ATE_IND.1	Pulse Secure will provide the TOE for testing.
AVA_VAN.1	Pulse Secure will provide the TOE for testing. Pulse Secure will provide a document identifying the list of software and hardware components.

Table 16 TOE Security Assurance Measures

6 TOE Summary Specification

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

TOE SFR	Rationale
FAU_GEN.1 FAU_GEN.2	The TSF generates audit records for the following events: • Startup and shutdown of the audit function • Administrative login and logout events • Changes to TSF data related to configuration changes • Generation of a CSR and associated keypair • Installation of a certificate • Resetting passwords • Low audit storage space available • Failure to establish a HTTPS/TLS session • Failure to establish a TLS session • All use of the identification and authentication mechanism (local and remote connections to the TSF) • Unsuccessful attempts to validate a certificate • Initiation of a software update • Result of a software update • Changes to the time • Modification of the behavior of the TSF • Failure of self-tests • Initiation and termination of the trusted channel • Initiation and termination of the trusted path • Attempts to unlock an interactive session • Termination of a session by the session locking mechanism Each audit record includes the date and time, type, subject identity (IP address, hostname, and/or username), the outcome (success or failure), and any additional information specified in column three of Table 14. Certificates are identified in the log by the Certificate DN. All generating/importing of changing or deleting of cryptographic keys relate to certificates. Public keys associated with certificates are identified by the certificate DN and the term 'public key'.
FAU_STG.1 FAU_STG_EXT.1	The TOE is a standalone TOE. By default, the TSF allocates 200 MB to local audit storage; however, the administrator can configure the file size, up to 500 MB. The TSF divides the local audit storage between two audit files. When the current audit file reaches capacity; the TSF overwrites the inactive log file (if present), creates a new log file, switches logging to the new log file, and generates an audit log indicating that a log file reached capacity. The TSF protects audit data from unauthorized modification and deletion though the restrictive administrative interfaces. The filesystem of the TSF is not exposed to the administrative user over the HTTPs GUI or the local CLI. The administrative user must be positively identified and authenticated prior to being allowed to clear the local audit log or change audit settings. The TSF implements Syslog over TLS using either TLS v1.1 or TLS v1.2. Logs are sent to the Syslog servers is real-time. The trusted channel with the Syslog server is described in greater detail in the FCS_TLSC_EXT.2 description.
FCS_CKM.1	The TSF supports the generation of RSA 2048 bit and 3072 bit keys for TLS client authentication, TLS server authentication, and RSA key encapsulation.

TOE SFR	Rationale
	The TSF generates ECDSA P-256 and P-384 keys for TLS client authentication, TLS server authentication, and TLS ECDHE key establishment.
FCS_CKM.2	The TSF uses both elliptic curve-based and RSA-based key establishment in support of TLS. When the TOE is configured with a server certificate with an RSA key, then RSA-based key establishment is used and the TOE acts as the sender. When the TOE is configured with a server certificate with an ECDSA key, then elliptic curve-based establishment is used and the TOE acts as the sender. The TOE supports the following schemes for key establishment: • RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1 • Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" For syslog server, the TSF acts as the client and is the recipient. For these sessions, the TSF utilizes elliptic curve key agreement when an ECDHE TLS ciphersuite is negotiated and RSA based key encapsulation when any other TLS ciphersuite is negotiated. See FCS_TLS* description below for more details.
FCS_CKM.4	The TSF stores the following persistent keys on internal Hard Disk Drives in plaintext: • HTTPs/TLS Private Host Key – generated using the DRBG and FCS_CKM.1 or entered by the Security Administrator. • Syslog/TLS Private Client Key – generated using the DRBG and FCS_CKM.1 or entered by the Security Administrator. The TSF stores loads the persistent keys into RAM when they are used and the TSF also stores the following ephemeral keys in RAM: • TLS Session keys – Established according to FCS_CKM.2 and derived using the TLS KDF • DRBG State – Derived from the entropy source The HTTPS/TLS Private Host Key and the Syslog/TLS Private Client key are zeroized from the disk when the Security Administrator deletes the key, replaces the key, or zeroizes the entire TOE. These keys are zeroized from RAM when the HTTP or Syslog process terminates. The TLS Session keys are zeroized from RAM when the associated TLS session is terminated. The DRBG state is zeroized when the TSF is shutdown or restarted. The TSF zeroizes keys in RAM by writing zeros to the memory location three times and performing a read verify to ensure that the memory location was set to all zeros. If the read verify fails, the TSF repeats the zeroization process. The TSF zeroizes the HTTPs/TLS Private Host Key and the Syslog/TLS Private Client key on the hard disk drives by overwriting the file location with data from /dev/urandom three

TOE SFR	Rationale
	times. Each overwrite calls /dev/urandom ensuring that a different pseudo random pattern is used each time. The above key destruction methods apply to all configurations and circumstances.
FCS_COP.1/DataEncryption	The TOE provides AES encryption/decryption in CBC and GCM modes with 128 and 256 bit keys.
FCS_COP.1/SigGen	The TOE supports signature generation and verification with RSA (2048-. 3072-bit) with SHA-1/256/384/512 in accordance with FIPS PUB 186-4 and ECDSA with NIST curves P-256 and P-384 with SHA-1/256/384/512 in accordance with FIPS PUB 196-4. These signatures support TLS authentication.
FCS_COP.1/Hash	The TOE provides cryptographic hashing services for key generation using SHA-256 as specified in NIST SP 800-90 DRBG. SHA-1, SHA-256, and SHA-384 are used in support of TLS. SHA-256 is used for file integrity checking and password obfuscation. SHA-512 is used for hashing of the digital signature to verify the firmware manifest file.
FCS_COP.1/KeyedHash	The TOE implements HMAC message authentication for the following uses: • TLSv1.1 Master Secret Derivation: HMAC-SHA1, key sizes of 128 bits with ECDH P-256 or 192 bits with RSA and ECDH P-384, block size 512 bits, and output length of 160 bits; • TLSv1.2 Master Secret Derivation: HMAC-SHA256, key sizes of 128 bits with ECDH P-256 or 192 bits with RSA and ECDH P-384, block size 512 bits, and output length of 256 bits; • TLSv1.2 Master Secret Derivation: HMAC-SHA384, key sizes of 256 bits with ECDH P-256 or 384 bits with RSA and ECDH P-384, block size 1024 bits, and output length of 384 bits; • TLSv1.1 Key Block Derivation: HMAC-SHA1, key size of 192 bits, block size of 512 bits, and output length of 160 bits; • TLSv1.2 Key Block Derivation: HMAC-SHA256, key size of 384 bits, block size of 512 bits, and output length of 256 bits; • TLSv1.2 Key Block Derivation: HMAC-SHA384, key size of 384 bits, block size of 1024 bits, and output length of 384 bits • TLS Message Authentication: HMAC-SHA1, key size of 160 bits, block size 512 bits, and output length of 160 bits
FCS_HTTPS_EXT.1 FCS_TLSS_EXT.1 FCS_TLSC_EXT.1 FCS_TLSC_EXT.2	The TSF implements the server and client sides of the HTTPs protocol according to RFC 2818 by using a TLS session to secure the HTTP session. All MUST and REQUIRED statement within RFC 2818 are followed. The TSF supports TLSv1.1 and TLSv1.2 for HTTPs/TLS. If the TSF receives a ClientHello message that requests TLSv1.0 or earlier, the TSF sends a fatal handshake_failure message and terminates the connection. When configured with an RSA certificate, the TSF supports the following TLS ciphersuties for connections to the TOE: • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256 • TLS_RSA_WITH_AES_256_CBC_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

TOE SFR	Rationale
	When configured with an ECDSA certificate, the TSF supports the following TLS ciphersuites for connections to the TOE: • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 The TOE conforms to RFC 5246, section 7.4.3 for key exchange. When the TSF selects an ECDHE ciphersuite, it sends the client secp256r1 or secp384r1 key agreement parameters. The TSF prefers secp256r1 if the client indicates support for both curves in the ClientHello message. The TSF implements a TLSv1.1 and TLSv1.2 client to secure communications with the Syslog server. The TSF supports and proposes the following ciphersuites and extensions in the ClientHello Message: • Ciphersuites for Syslog communications (FCS_TLSC_EXT.2): ○ TLS_RSA_WITH_AES_128_CBC_SHA ○ TLS_RSA_WITH_AES_256_CBC_SHA ○ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ○ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA ○ TLS_RSA_WITH_AES_128_CBC_SHA256 ○ TLS_RSA_WITH_AES_256_CBC_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 ○ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 ○ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ○ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • Signature Algorithms: ○ RSA with SHA-1/256/384/512 ○ ECDSA SHA-1/256/384 • Supported Elliptic Curves: ○ secp256r1 ○ secp384r1 • Supported Point Formats ○ Uncompressed The Security Administrator can configure the TSF so it only accepts TLSv1.2 connections for server and client communications. The Security Administrator can enable and disable individual ciphersuites as well as specifying the preferred ordering of ciphersuites. The TOE sends the supported elliptic curves extension if an ECDHE ciphersuite is selected and does not require administrator intervention. When the Syslog server sends the Certificate Request message, the TSF replies with a Client Certificate message. The Client Certificate message includes the certificate that the

TOE SFR	Rationale
	Security Administrator configured to authenticate to the Syslog server. The TSF establishes reference identifiers for the remote server as follows: • When the server is specified using a domain name, the TSF verifies that the domain name matches a Subject Alternative Name DNS Name field in the certificate using exact or wildcard matching specified in Section 3.1 of RFC 2818. If the certificate does not contain any Subject Alternative Name fields, the TSF matches the domain name against the Common Name in the certificate. • When the server is specified using an IP address, the TSF verifies that the IP address exactly matches a Subject Alterative Name IP Address field in the certificate using the rules specified in Section 3.1 of RFC 2818. If the certificate does not contain any Subject Alternative Name fields, the TSF matches the IP address against the Common Name in the certificate. The TSF does support wildcards but does not support certificate pinning and determines if the certificate is valid for the specified server based on the DNS name or IP address of the server. Wildcards are supported only at the left-most label of the identifier. In either instance, the TSF will not establish the connection if the peer certificate does not successfully authenticate the peer according to X.509 authentication. When acting as a server, the TSF listens on port 443 for HTTPs connections. The TSF uses HTML over HTTPs to present the administrative users with a secure management interface. The TSF uses TLS to provide a secure connection between the TSF and remote Security Administrators.
FCS_RBG_EXT.1	The TOE implements a DRBG in accordance with ISO/IEC 18031:2011 using a CTR DRBG with AES. The TSF seed the CTR_DRBG using 256-bits of data that contains at least 256 bits of entropy. The TSF gathers and pools entropy from 1 software (CPU jitter) and 1 hardware noise source (hard disk interrupts). The entropy sources are discussed in greater detail in the Entropy documentation.
FIA_AFL.1	An administrator can configure the number of unsuccessful attempts a remote administrator can make before a lock-out and can configure the length of time that the remote administrator is locked out. The attempts can range between 2 and 2147483647. The length of time can be configured between 1 and 10080 minutes. Additionally, the number of attempts can be configured per minute, i.e. that 3 failures within a minute result in a lockout. If the user enters an incorrect password the configured number of times, the user is locked out they cannot login through any remote interface on the TOE. When the lockout time has expired, the administrator is allowed to authenticate to the TOE again. Lockouts are not enforced on the TOE's console interface. This ensures that authentication failures cannot lead to a situation where no administrator access is available.
FIA_PMG_EXT.1	The TSF 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 printable ASCII characters (values 0x20 – 0x7E)with a minimum length settable by the administrator and support 15 characters.
FIA_UIA_EXT.1 FIA_UAU_EXT.2	The TSF utilizes HTTPS to secure a remote administration web UI session. When connecting over HTTPS, the TSF presents Security Administrators with a username and

TOE SFR	Rationale
	password prompt; however, the Security Administrator can choose to authenticate using an X.509 certificate. In this case, the TSF forces a TLS Renegotiation. The TSF includes the Certificate Request message as part of the handshake so the client will send a Client Certificate message to authenticate the Security Administrator. The Security Administrator using password authentication is considered authenticated if the username and the SHA-256 hash of the password matches the stored username and SHA-256 password hash. A successful authentication takes the user to the System Status page. The Security Administrator using certificate authentication is considered authenticated if the presented certificate's DN and public key match the DN and public key stored on the TOE and associated with the user's identity. The TSF utilizes a local serial CLI which presents Security Administrators with a username and password prompt. The Security Administrator is considered authenticated if the username and password provided match the credentials configured in the TSF. A successful login takes the user to the CLI menu. Prior to successful identification and authentication, the TSF displays the TOE access banner specified in FTA_TAB.1 and responds to ICMP Echo messages with ICMP Echo Reply messages.
FIA_UAU.7	When the user is entering their password over the local console, the TSF does not echo any characters back.
FIA_X509_EXT.1/Rev	When a certificate is used (to identify the TSF or identify an external entity to the TSF), the TSF verifies certificates by checking the following: 1. The current date between the "Valid from" and "Valid to" dates. 2. The certificate is not listed on the CRL. If the TSF has a cached response that has not expired, the TSF uses the cached response in lieu of querying the CRL server. 3. The certificate chain is valid: • Each certificate in the certificate chain passes the checks described in #1 and #2. • Each certificate (other than the first certificate) in the certificate chain has the Subject Type=CA flag set. • Each certificate is signed by: o a certificate in the certificate chain, or o a trusted root CA that has been installed in the TSF The TSF verifies the validity of a certificate when: • An HTTPS client establishes a TLS connection (HTTPs Server Certificate) • An HTTPS client presents a client authentication certificate • The TSF verifies the server certificate of the Syslog server • The TSF uses its client certificate authenticate to the Syslog server If the Security Administrator loads a certificate with a Subject Type=CA, the TSF does not validate the certificate path. The rules for extendedKeyUsage fields are followed in all instances.
FIA_X509_EXT.2	The when establishing a connection to the Syslog server, the TSF uses the certificate presented by the Syslog server to verify the server's identity. The TSF establishes reference identifiers for the remote server as follows:

TOE SFR	Rationale
	• When the server is specified using a domain name, the TSF verifies that the domain name matches a Subject Alternative Name DNS Name field in the certificate using exact or wildcard matching specified in Section 3.1 of RFC 2818. If the certificate does not contain any Subject Alternative Name fields, the TSF matches the domain name against the Common Name in the certificate. • When the server is specified using an IP address, the TSF verifies that the IP address exactly matches a Subject Alterative Name IP Address field in the certificate using the rules specified in Section 3.1 of RFC 2818. If the certificate does not contain any Subject Alternative Name fields, the TSF matches the IP address against the Common Name in the certificate. Once the TSF has verified that the certificate identifiers are valid for the Syslog server, the TSF verifies the validity of the certificate as described in FIA_X509_EXT.1/Rev. The TSF presents its own certificate to the Syslog server. This certificate is configured specifically for authentication to the Syslog server by the Security Administrator. When a user connects over HTTPS, the TSF presents the certificate that the Security Administrator configured for use with HTTPS. If the user proceeds with certificate authentication, the TSF verifies that the certificate presented by the user is the same certificate that is configured for the provided username. If the TSF cannot contact the CRL server or the server does not respond, the TSF logs the failure and considers the certificate valid. If any of the other FIA_X509_EXT.1/Rev validity checks fail, the TSF rejects the certificate and does not establish the connection.
FIA_X509_EXT.3	The TSF allows Security Administrators to generate Certificate Signing Requests. The TSF requires the Security Administrator to specify the following values: • Common Name • Organization • Locality • State • Country • Key Type (RSA or ECDSA) • Key Length (2048, 3072, P-256, or P-384) The TSF allows the Security Administrator to specify an Organization Unit and additional random data used when generating the key pair. This information is optional/not required for creating Certificate Signing Requests.
FMT_MOF.1/Functions FMT_MOF.1/ManualUpdate FMT_MTD.1/CoreData FMT_MTD.1/CryptoKeys FMT_SMF.1 FMT_SMR.2	The TSF implements the Security Administrator role to authorized administrators of the TOE. The TSF allows the Security Administrators to administer the TSF via CLI through a serial cabled connected to the TOE and a web UI over a remote HTTPS channel. The TSF permissions restrict access to these management functions to users that have been identified, authenticated, and authorized with the Security Administrator role. The web UI and local console allow the Security Administrator to perform the following TSF management functions: • Verify/Install Firmware Updates • View/Edit settings for sending audit data to the Syslog Server • View/Edit the amount of space allocated Local Audit storage • Clear/Delete Local Audit records • View/Edit enabled TLS versions

TOE SFR	Rationale
	• View/Edit enabled TLS ciphersuties • View/Edit X.509 Certificates • Generate and configure cryptographic keys used to identify the TOE • Configure cryptographic keys used to authenticate users • View/Edit the TOE access banner • View/Edit the session inactivity timeout • View/Edit authentication failure parameters • Set user account passwords • Modify system time The administrative interfaces provided by the TSF do not allow any of these functions to be accessed by unauthenticated or unauthorized users. The TOE provides a trust store to store certificates. The permissions on the trust store restrict access so that only Security Administrators can import or delete certificates from the trust store. Security Administrators can also view the certificates stored in the trust store. No other access to the trust store is allowed. The only functions accessible prior to authentication are the display of the configurable warning and consent banner and the automated response to ICMP echo messages with ICMP echo reply messages.
FPT_SKP_EXT.1	The TSF stores pre-shared keys, symmetric keys, and private keys in plaintext on the hard disk; however, it does not provide an interface to allow any user to view any of these values.
FPT_APW_EXT.1	The TSF does not store plaintext password. The TSF stores the SHA-256 hash of each users' password. Additionally, the TSF does not provide a user interface to view the password hashes.

FPT_TST_EXT.1	The TSF performs the following hardware self-tests at power-on:
	• BIOS checks at power-on (on hardware platforms only)
	  ○ 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.
	• Cryptographic library tests:
	  ○ HMAC-SHA-256 integrity check of the library
	  ○ HMAC-SHA-1 KAT
	  ○ HMAC-SHA-256 KAT
	  ○ HMAC-SHA-384 KAT
	  ○ AES 128 ECB Encrypt and Decrypt KAT
	  ○ AES 256 GCM Encrypt and Decrypt KAT
	  ○ RSA 2048 SHA-256 Sign and Verify KAT
	  ○ ECDSA P-224 SHA-512 Sign and Verify PCT
	  ○ DRBG AES-CTR-256 KAT (invoking the instantiate, reseed, and generate functions)
	• Firmware checks:
	  ○ RSA 2048 SHA-512 digital signature verification of the manifest file. This file contains a list of all executables that are part of the TSF

TOE SFR	Rationale
	○ SHA-256 integrity check of each executable file in the TSF using the pre-calculated hashes from the manifest file. The BIOS checks the successful use of the hardware platform to perform cryptographic operations and provides basic assurance that the hardware is working properly. The Cryptographic library test and the Firmware checks provide a high level of assurance that the firmware has not been tampered with and that the cryptographic algorithms are working properly. The Cryptographic library tests verify that each cryptographic algorithm[1] specified in FCS_COP.1 requirements is passing a KAT. The KAT demonstrates that the algorithm is functioning properly by invoking the algorithm with hard coded keys and messages and comparing the result to a pre-computed, known to be correct value. The ECDSA PCT shows that the ECDSA algorithm is functioning properly by signing a known value with a known key and verifying that verifying the computed signature indicates that the signature is valid. If the BIOS checks fail, the TSF does not power-up. If the cryptographic library tests fail, the TSF will not start up. If any of the other checks fail, the TSF will log the failure and continue to boot.
FPT_TUD_EXT.1	The TSF allows the Security Administrator to install firmware updates. The Security Administrator obtains candidate updates by downloading them from the Pulse Secure website. When the Security Administrator uploads a firmware update, the TSF performs an RSA 2048 SHA-256 digital signature verification of the update using the Pulse Secure firmware update public key. Pulse Secure retains control over the private key used to sign firmware updates. If the signature check is successful, the TSF installs the update. If the signature check detects tampering with the update and/or signature, the TSF presents the user with an error message and discards the update. The TSF allows the Security Administrator to view the currently running version of firmware from the System Maintenance > Platform page of the web UI.
FPT_STM_EXT.1	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 that is updated by a Security Administrator once a month. The TSF uses system time to timestamp audit log records, to determine user session timeouts, and to determine certificate validity. These uses of time do not require an accuracy finer that one second, and the frequency of updating the time keeps the clock drift under one second.
FTA_SSL_EXT.1 FTA_SSL.3 FTA_SSL.4	User sessions can be terminated by users. The Security Administrator can set the TOE so that local and remote sessions are terminated after a Security Administrator-configured period of inactivity.

[1] The TSF only tests a single set of parameters for each cryptographic algorithm.

TOE SFR	Rationale
FTA_TAB.1	The TSF enables Security 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 Security Administrator wishes to communicate. The TSF presents the access banner prior to authentication when a user connects to the remote web UI or local console CLI described in the FIA_UIA_EXT.1, FIA_UAU_EXT.2 description.
FTP_ITC.1	The TSF communicates with the external syslog server using Syslog over TLS with Authentication as described in the descriptions of FAU_STG_EXT.1 and FCS_TLSC_EXT.2. The TSF initiates the trusted channel with the Syslog server.
FTP_TRP.1/Admin	The TSF provides a trusted path for remote administration using HTTPs/TLS as described in FCS_HTTPS_EXT.1 and FCS_TLSS_EXT.1 descriptions.

Table 17 TOE Summary Specification SFR Description

7 Terms and Definitions

Abbreviations/Acronyms	Description
AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
BIOS	Basic Input/Output System
CBC	Cipher Block Chaining
CLI	Command Line Interface
CMOS	Memory used to store BIOS settings
CRL	Certificate Revocation List
CSR	Certificate Signing Request
DH	Diffie-Hellman
DHE	Diffie-Hellman Ephemeral
DMI	Device Management Interface
DNS	Domain Name System
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
ECDH	Elliptic Curve Diffie-Hellman
ECDHE	Elliptic Curve Diffie-Hellman Ephemeral
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
GUI	Graphical User Interface
HMAC	Hash Message Authentication Code
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPs	Hypertext Transfer Protocol Secure
IP	Internet Protocol
KAT	Known Answer Test
KDF	Key Derivation Function
MB	Megabyte
NAC	Network Access Control
PCT	Pairwise Consistency Test
PKCS	Public Key Cryptography Standards
RAM	Random Access Memory
RFC	Requests for Comments
RSA	Rivest-Shamir-Adleman
SAML	Security Assertion Markup Language
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSO	Single Sign On
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UI	User Interface
URI	Uniform Resource Identifier
VPN	Virtual Private Network

Table 18 TOE Abbreviations and Acronyms

Abbreviations/Acronyms	Description
CC	Common Criteria
CCRA	Arrangement on the Recognition of Common Criteria Certificates in the field of IT Security
DOD	Department of Defense
NIAP	National Information Assurance Partnership
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
TRRT	Technical Rapid Response Team
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification

Table 19 CC Abbreviations and Acronyms

---End of Document---