

Palo Alto Networks Cortex XSOAR Engine 6.14 Security Target

Version: 0.3
Date: September 16, 2026



XSOAR

Palo Alto Networks, Inc.
www.paloaltonetworks.com
© 2026 Palo Alto Networks, Inc. Palo Alto Networks is a registered trademark of Palo Alto Networks. A list of our trademarks can be found at https://www.paloaltonetworks.com/company/trademarks.html. All other marks mentioned herein may be trademarks of their respective companies.

Table of Contents

1. SECURITY TARGET INTRODUCTION 4
1.1 SECURITY TARGET, TOE AND CC IDENTIFICATION 4
1.2 CONFORMANCE CLAIMS 5
1.3 CONVENTIONS 5
1.3.1 Terminology 6
1.3.2 Acronyms 6
2. PRODUCT DESCRIPTION 8
2.1 TOE OVERVIEW 8
2.2 TOE ARCHITECTURE 9
2.2.1 Physical Boundaries 9
2.2.2 Logical Boundaries 10
2.3 TOE DOCUMENTATION 11
3. SECURITY PROBLEM DEFINITION 12
4. SECURITY OBJECTIVES 13
4.1 SECURITY OBJECTIVES FOR THE TOE 13
4.2 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT 13
5. IT SECURITY REQUIREMENTS 14
5.1 EXTENDED REQUIREMENTS 14
5.2 TOE SECURITY FUNCTIONAL REQUIREMENTS 14
5.2.1 Cryptographic Support (FCS) 16
5.2.2 User Data Protection (FDP) 21
5.2.3 Identification and Authentication (FIA) 22
5.2.4 Security Management (FMT) 24
5.2.5 Privacy (FPR) 25
5.2.6 Protection of the TSF (FPT) 25
5.2.7 Trusted Path/Channel (FTP) 27
5.3 TOE SECURITY ASSURANCE REQUIREMENTS 28
6. TOE SUMMARY SPECIFICATION 29
6.1 CRYPTOGRAPHIC SUPPORT 29
6.2 USER DATA PROTECTION 31
6.3 IDENTIFICATION AND AUTHENTICATION 31
6.4 SECURITY MANAGEMENT 33
6.5 PRIVACY 33
6.6 PROTECTION OF THE TSF 33
6.7 TRUSTED PATH/CHANNEL 35
7. PROTECTION PROFILE CLAIMS 36
8. RATIONALE 37

LIST OF FIGURES

Figure 1: TOE Deployment 8
Figure 2: TOE Architecture 9

LIST OF TABLES

Table 1 TOE Security Functional Components 14
Table 2 Assurance Components 28
Table 3 Cryptographic Functions and CAVP Certificates 30

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is the Cortex XSOAR Engine version 6.14.

The Palo Alto Networks Cortex XSOAR (eXtended Security Orchestration, Automation and Response) combines security orchestration, threat intel and incident management, and automated investigation.

The focus on this evaluation is on the TOE functionality supporting the claims in the Protection Profile for Application Software.

The Security Target contains the following additional sections:

- Product Description (Section 2)
- Security Problem Definition (Section 3)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8).

1.1 Security Target, TOE and CC Identification

ST Title – Palo Alto Networks Cortex XSOAR Engine 6.14 Security Target

ST Version – Version 0.3

ST Date – September 16, 2026

TOE Identification –

- Cortex XSOAR Engine 6.14.0 running on Ubuntu 24.04[1].

[1] While the TOE was tested on Ubuntu 24.04, additional Linux environments such as RHEL 8/9/10, Ubuntu (20.04, 22.04), Oracle Linux 8.9/9.3/9.4, Amazon Linux 2/2023, and Rocky Linux 9.5/9.6 are supported as well. This is vendor affirmed with ACVP algorithm testing. The minimum hardware requirements on those systems are 16 CPU cores, 32 GB of memory, and 1 TB SSD hard disk.

TOE Developer – Palo Alto Networks, Inc.

Evaluation Sponsor – Palo Alto Networks, Inc.

CC Identification – *Common Criteria for Information Technology Security Evaluation, Common Criteria CC:2022, Revision 1*

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications: This ST is conformant to:

- Protection Profile for Application Software, Version 2.0, June 16, 2025 [APPSW].
- Functional Package for Transport Layer Security (TLS), Version 2.1, August 25, 2025 [PKGTLS]
- Functional Package for X.509, Version 1.0, October 21, 2024 [PKGX509]

This TOE and ST are conformant to Parts 2 (extended) and 3 (extended) of Common Criteria CC:2022, Revision 1.

The following NIAP Technical Decision(s) apply to this APPSW, and have been accounted for in the ST development:

- TD0994 Addition of FCS_COP.1/XOF to PP_APP_V2.0
- TD1003 Adding MOD_EDR V2.0 and MOD_HA to the Conformance Claims for PP_APP_V2.0
- TD1026 Clarifications to FMT_MEC_EXT.1 Windows Test
- TD1050 Clarification to App Note for FPT_LIB_EXT.1.1

The TOE and ST are package-name conformant to [PKGTLS and PKGX509].

The following NIAP Technical Decision(s) apply to PKGTLS or PKGX509, and have been accounted for in the ST development:

- TD0969 Correction to Session Resumption Test
- TD0970 Clarification to Selections in FIA_X509_EXT.1.4
- TD0977 Removal of Deplicative Test Case in FIA_X509_EXT.1
- TD0979 Correction to Section 1.3 FCS_RBG.1 Entry
- TD0980 Correction to FCS_(D)TLSS_EXT.1.4 and FCS_(D)TLSC_EXT.1.4
- TD0981 Correction to Test FCS_TLSS_EXT.1.1:5.3.5
- TD0989 Correction to FCS_TLSC_EXT.1.1:8.1 in PKCS_TLS_V2.1
- TD0993 Administrative Corrections for PKG_X509_V1.0
- TD1008 (D)TLS(C/S) Corrections
- TD1025 Restriction to CNSA-Compliant Algorithms for (D)TLS Servers
- TD1038 Corrections to Application Notes Regarding Extended Master Secret Extension
- TD1057 Clarification RE Signing in FIA_X509_EXT.1.1

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.

- Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a number in parentheses placed at the end of the component. For example, FDP_ACC.1 (1) and FDP_ACC.1 (2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement, (1) and (2).
- Assignment: allows the specification of an identified parameter. Assignments are indicated using italicized and are surrounded by brackets (e.g., [*assignment*]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
- Selection: allows the specification of one or more elements from a list. Selections are indicated using bold and are surrounded by brackets (e.g., [**selection**]).
- Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… some **big** things …"). Note that 'cases' that are not applicable in a given SFR have simply been removed without any explicit identification.

- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.
- The ST does not highlight operations that have been completed by the PP and EP authors.

1.3.1 Terminology

The following terms and abbreviations are used in this ST:

Playbooks	Playbooks are at the heart of the Cortex XSOAR system. They enable you to automate many of your security processes, including, but not limited to handling your investigations and managing your tickets.

1.3.2 Acronyms

AES	Advanced Encryption Standard
CBC	Cipher-Block Chaining
CC	Common Criteria for Information Technology Security Evaluation
CEM	Common Evaluation Methodology for Information Technology Security
CM	Configuration Management
CLI	Command Line Interface
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
FIPS	Federal Information Processing Standard
FSP	Functional Specification
GCM	Galois/Counter Mode
GUI	Graphical User Interface
HMAC	Hashed Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
NIST	National Institute of Standards and Technology
PP	Protection Profile
RHEL	Red Hat Enterprise Linux

RSA	Rivest, Shamir and Adleman (algorithm for public-key cryptography)
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SOAR	Security Orchestration, Automation and Response
SOC	Security Operation Center
SSH	Secure Shell
SSL	Secure Socket Layer
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functions
URL	Uniform Resource Locator
VM	Virtual Machine

2. Product Description

The Target of Evaluation (TOE) is the Palo Alto Networks Cortex[2] XSOAR Engine 6.14. Cortex XSOAR combines security orchestration, incident management, and interactive investigation into a seamless experience. The orchestration component is designed to automate security product tasks and weave in human analyst tasks and workflows.

2.1 TOE Overview

The TOE is a software application that runs on various Linux environments (server/workstation) and works with a XSOAR Server in the operational environment to provide SOC team with detection and automatic response capabilities[3]. The Server (in the operational environment) provides UI functionality and playbook detection/response functionality, while Engine (as known as the TOE) is used to efficiently share the workload (e.g., load-balancing), thereby speeding up execution time. In essence, the TOE is like an extension of the Server used to offload tasks to it.

The Server (in the operational environment) provides security management functions and interface via web UI protected by HTTPS (out of scope) and communicates with the TOE through TLSv1.2 protected channels. The Server implements TLS server functionality while the TOE implements TLS client functionality. The Server and TOE can be deployed on the same workstation/server, or they can be deployed on separate devices.

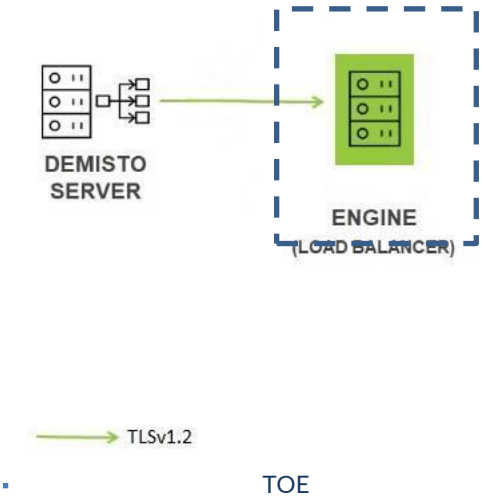


Figure 1: TOE

[2] Cortex was formerly known as Demisto.
[3] These capabilities are out of the evaluation scope and are provided for informational purposes only.

FIPS-CC Mode of Operation

The FIPS-CC Version of Cortex XSOAR are, by default, compliant with all FIPS and CC Approved algorithms, protocols, and key sizes requirements. The FIPS-CC mode is enabled by default and cannot be disabled. Additional configurations must be performed to enable X509v3 certification validation, revocation checking, and mutual authentication. All configuration steps are fully documented in the CC Evaluated Configuration Guide (also known as the AGD).

2.2 TOE Architecture

The TOE is a software solution that is comprised of the items noted in Section 2.2.1. The software is available for download from the Palo Alto Networks support site.

The following diagram depicts the software architecture of the TOE.

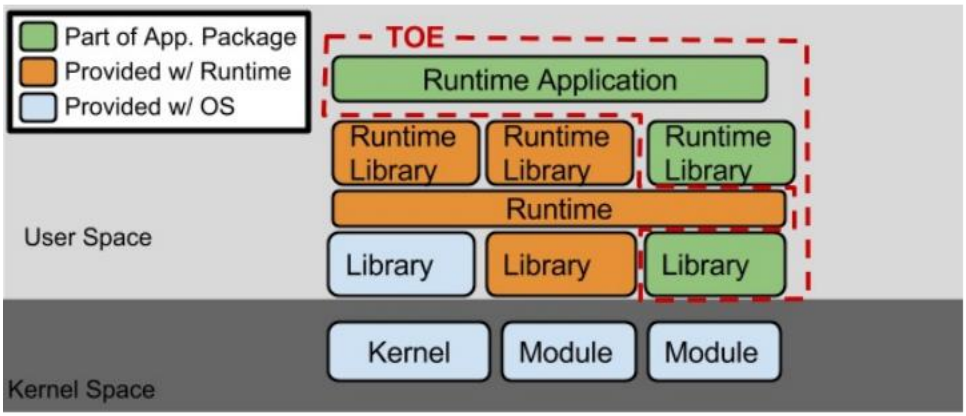


Figure 2: TOE Architecture

2.2.1 Physical Boundaries

The physical boundary of the TOE is the application installed and running on a supported platform (i.e., Linux operating systems).

2.2.1.1 Software Requirements

The TOE runs on an operating system that includes RHEL 8/9/10, Ubuntu (20.04, 22.04, 24.04), Oracle Linux 8.9/9.3/9.4, Amazon Linux 2/2023, or Rocky Linux 9.5/9.6.

2.2.1.2 Hardware Requirements

The TOE must be installed on the system with the minimum hardware requirements specified below.

Minimum hardware: 16 CPU (minimum); 32 GB RAM (minimum); 1 TB of disk space (minimum)

The TOE was installed and tested on the following platforms:

- Ubuntu 24.04 - Processor: Intel Xeon Gold 6248 (Cascade Lake)

2.2.2 Logical Boundaries

This section summarizes the security functions provided by the TOE:

- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Privacy
- Protection of the TSF
- Trusted Path/Channels

2.2.2.1 Cryptographic Support

The TOE implements CAVP validated cryptographic algorithms that provide key management, random bit generation, encryption/decryption, digital signature and cryptographic hashing and keyed-hash message authentication features in support of cryptographic protocols such as TLS.

2.2.2.2 User Data Protection

The TOE provides no access to hardware resources or sensitive information repositories. The TOE restricts network communication to application-initiated communication between the TOE and Server, and stores no sensitive data.

2.2.2.3 Identification and Authentication

The TOE authenticates using X509v3 certificate-based method.

2.2.2.4 Security Management

The TOE provides access to the security management functions via configuration files. Identification and authentication are required by the operating system before accessing the files. In addition, the operating system can provide some configuration options for the TOE. In that case, the operating system I&A method and privileges will be used and enforced.

2.2.2.5 Privacy

The TOE does not transmit PII over the network.

2.2.2.6 Protection of the TSF

The TOE implements a number of functions to ensure that it is protected against tampering and corruption. These mechanisms include utilizing platform APIs, memory mapping, and stack-based buffer overflow protection. The TOE is implemented in Golang which is designed as a memory safe language. This includes automatic bounds checking, garbage collection, and type safety check. Palo Alto Networks provides customers with a means of updating their TOE using trusted updates. These trusted updates (signed DEB package) are securely delivered over HTTPS website and verified using approved digital signature methods. All of these updates are properly signed and are verified by the operating system mechanism. In addition, the TOE image is protected with FIPS Software integrity test at power-up (e.g., when the application is started or is reloaded).

2.2.2.7 Trusted Path/Channels

The TOE protects communication with Server, in the operational environment, using TLS to ensure both integrity and disclosure protection.

2.3 TOE Documentation

Palo Alto Networks Inc. offers a series of documents that describe the installation of Palo Alto Networks Cortex XSOAR as well as guidance for subsequent use and administration of the applicable security features.

For Cortex XSOAR v6.14, these documents include:

- Palo Alto Networks Common Criteria Evaluated Configuration Guide (CCECG) for Cortex XSOAR Server and Engine 6.14, 07/15/2026

3. Security Problem Definition

This security target includes by reference the Security Problem Definition (composed of organizational policies, threat statements, and assumption) from [APPSW].

In general, the [APPSW] has presented a Security Problem Definition appropriate for software applications, and as such, is applicable to the TOE.

The following threats are directly from the [APPSW]:

T. NETWORK_ATTACK	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with the application software or alter communications between the application software and other endpoints in order to compromise it.
T.NETWORK_EAVESDROP	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between the application and other endpoints.
T.LOCAL_ATTACK	An attacker can act through unprivileged software on the same computing platform on which the application executes. Attackers may provide maliciously formatted input to the application in the form of files or other local communications.
T.PHYSICAL_ACCESS	An attacker may try to access sensitive data at rest.

The following assumptions are made as drawn directly from the [APPSW]:

A. PLATFORM	The TOE relies upon a trustworthy computing platform with a reliable time clock for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.
A. PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy.
A. PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software in compliance with the applied enterprise security policy.

.

4. Security Objectives

The sections below identify the security objectives for the TOE and for the operational environment. These security objectives identify the responsibilities of the TOE and the operational environment in meeting security needs.

4.1 Security Objectives for the TOE

See Section 4 of [APPSW].

4.2 Security Objectives for the Operational Environment

OE.PLATFORM	The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying operating system and any discrete execution environment provided to the TOE.
OE.PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy.
OE.PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

5. IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the following Protection Profiles (PP):

- *Protection Profile for Application Software, Version 2.0, 16 June 2025* [APPSW],
- *Functional Package for Transport Layer Security (TLS), Version 2.1, 25 August 2025 [PKGTLS],*
- *Functional Package for X.509, Version 1.0, 21 October 2024 [PKGX509]*

The SARs are the set of SARs specified in [APPSW].

5.1 Extended Requirements

All extended requirements in this ST have been drawn from the [APPSW], [PKGTLS], and [PKGX509]. The [APPSW], [PKGTLS], and [PKGX509] define all the extended SFRs (*_EXT.1) and since they are not redefined in this ST, the [APPSW], [PKGSTLS], and [PKGX509] should be consulted for more information in regard to those CC extensions.

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the Palo Alto TOE.

Table 1 TOE Security Functional Components

Requirement Class	Requirement Component
FCS: Cryptographic Support	FCS_CKM_EXT.1 Cryptographic Key Generation Services
	FCS_CKM.1/AK Cryptographic Asymmetric Key Generation
	FCS_CKM.1/SK Cryptographic Symmetric Key Generation
	FCS_CKM.2 Cryptographic Key Establishment
	FCS_COP.1/SKC Cryptographic Operation – Encryption/Decryption
	FCS_COP.1/Hash Cryptographic Operation – Hashing
	FCS_COP.1/KeyedHash Cryptographic Operation – Keyed-Hash Message Authentication
	FCS_COP.1/SigGen Cryptographic Operation - Signing Generation
	FCS_COP.1/SigVer Cryptographic Operation - Signing Verification
	FCS_RBG_EXT.1 Random Bit Generation Services
	FCS_RBG.1 Random Bit Generation (RBG)
	FCS_RBG.2 Random Bit Generation (External Seeding)
	FCS_STO_EXT.1 Storage of Credentials
	FCS_TLS_EXT.1 TLS Protocol
	FCS_TLSC_EXT.1 TLS Client Protocol
	FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

Requirement Class	Requirement Component
	FCS_TLSC_EXT.4 TLS Client Support for Renegotiation
FDP: User Data Protection	FDP_DEC_EXT.1 Access to Platform Resources
	FDP_NET_EXT.1 Network Communications
	FDP_DAR_EXT.1 Encryption of Sensitive Application Data
FIA: Identification and Authentication	FIA_X509_EXT.1 X.509 Certificate Validation
	FIA_X509_EXT.2 X.509 Certificate Support for Functions
	FIA_X509_EXT.3 X.509 Certificate Requests
	FIA_XCU_EXT.1 Implementation of X.509 Functions
	FIA_XCU_EXT.2 X.509 Certificate Acquisition
FMT: Security Management	FMT_MEC_EXT.1 Supported Configuration Mechanism
	FMT_CFG_EXT.1 Secure by Default Configuration
	FMT_SMF.1 Specification of Management Functions
FPR: Privacy	FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information
FPT: Protection of the TSF	FPT_API_EXT.1 Use of Supported Services and APIs
	FPT_AEX_EXT.1 Anti-Exploitation Capabilities
	FPT_FLS.1 Failure with Preservation of Secure State
	FPT_TST.1 TSF Self-Testing
	FPT_TUD_EXT.1 Support for Trusted Updates
	FPT_TUD_EXT.2 Integrity for Installation and Update
	FPT_LIB_EXT.1 Use of Third Party Libraries
FTP: Trusted Path/Channel	FTP_DIT_EXT.1 Protection of Data in Transit

5.2.1 Cryptographic Support (FCS)

FCS_CKM_EXT.1 – Cryptographic Key Generation Services

FCS_CKM_EXT.1.1 The application shall [

- ***implement asymmetric key generation***

].

FCS_CKM.1/AK – Cryptographic Asymmetric Key Generation

FCS_CKM.1.1/AK The **application** shall **[**

- ***implement functionality***

] to generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm [

- ***CNSA 1.0 Compliant Algorithms:***
- *[**ECC schemes**] using [**"NIST curves" P-384 and [P-521]** that meet the following: **[FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2]***

].

FCS_CKM.1/SK – Cryptographic Symmetric Key Generation

FCS_CKM.1.1/SK The application shall **[*implement functionality*]** to generate **symmetric** cryptographic key **using a Random Bit Generator as specified in FCS_RBG_EXT.1** and specified cryptographic key sizes *256-bit*

FCS_CKM.2 – Cryptographic Key Establishment

FCS_CKM.2.1 The application shall [***implement functionality***] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:

[

- ***CNSA 1.0 Compliant Algorithms:***
  - ***[Elliptical curve-based key establishment schemes] that meets the following: [NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"]***

].

FCS_COP.1/SKC – Cryptographic Operation – Encryption/Decryption

FCS_COP.1.1/SKC The **application** shall [***perform***] [*encryption/decryption*] in accordance with a specified cryptographic algorithm **[**

- ***AES-GCM (as defined in NIST SP 800-38D) mode***

] and cryptographic key sizes [***256-bits***]**.**

FCS_COP.1/Hash – Cryptographic Operation – Hashing

FCS_COP.1.1/Hash The **application** shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [

- ***SHA-256,***
- ***SHA-384***
- ***SHA-512***

] and **message digest** sizes [

- ***256,***
- ***384***
- ***512***

] **bits** that meet the following: [*FIPS Pub 180-4, "Secure Hash Standard"*].

Application Note*: Support for SHA-1 is needed for digital signature verification only. The TOE must hash with SHA-256 or higher.*

FCS_COP.1/KeyedHash – Cryptographic Operation – Keyed-Hash Message Authentication

FCS_COP.1.1/KeyedHash The **application** shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm [

- ***HMAC-SHA-384,***
- ***HMAC-SHA-512***]

] **with** key sizes [*384, 512*] **and message digest sizes** [***384, 512***] **bits** that meet the following: [*FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code," and FIPS Pub 180-4, "Secure Hash Standard"*]**.**

FCS_COP.1/SigGen – Cryptographic Operation – Signing Generation

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

- ***CNSA 1.0 Compliant Algorithms: [***

- ***RSA schemes*** *using cryptographic key sizes of [3072-bit or greater] that meet the following: [FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5]*
- ***ECDSA schemes*** *using "NIST curves" P-256, P-384 and [**no other curves**] that meet the following: [FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6]*

]

].

FCS_COP.1/SigVer – Cryptographic Operation – Signing Verification

FCS_COP.1.1/SigVer The **application** shall perform [*cryptographic signature services (verification)*] in accordance with a specified cryptographic algorithm [

- ***CNSA 1.0 Compliant Algorithms: [***
  - ***RSA schemes*** *using cryptographic key sizes of [3072-bit or greater] that meet the following: [FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5]*
  - ***ECDSA schemes*** *using "NIST curves" [**P-384, P-521**] that meet the following: [FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6]*

]

].

FCS_RBG_EXT.1 – Random Bit Generation Services

FCS_RBG_EXT.1.1 The application shall [

- ***implement DRBG functionality***

] for its cryptographic operations.

FCS_RBG.1 – Random Bit Generation (RBG)

FCS_RBG.1.1 The TSF shall perform deterministic random bit generation services using [

- ***CTR_DRBG (AES)***

] in accordance with [NIST SP 800-90A] after initialization with a seed.

FCS_RBG.1.2 The TSF shall use a [***TSF interface for seeding***] for initialized seeding.

FCS_RBG.1.3 The TSF shall update the RBG state by [***reseeding***] using a [***TSF interface for seeding***] in the following situations: [

- ***on the condition: [platform is rebooted or every 4096 calls to RAND_bytes]***

] in accordance with [*FIPS 140-3 and SP 800-90A*].

FCS_RBG.2 – Random Bit Generation (External Seeding)

FCS_RBG.2.1 The TSF shall be able to accept a minimum input of [*384 bits*] from a TSF interface for the purpose of seeding.

FCS_STO_EXT.1 – Storage of Credentials

FCS_STO_EXT.1.1 The application shall [

- ***not store any credentials***

] to non-volatile memory.

FCS_TLS_EXT.1 – TLS Protocol

FCS_TLS_EXT.1 The TSF shall implement [

- ***TLS as a client***

].

FCS_TLSC_EXT.1 - TLS Client Protocol

FCS_TLSC_EXT.1.1 The TSF shall implement [***TLS 1.2 (RFC 5246)***] as a client that supports additional functionality for session renegotiation protection and [

- ***mutual authentication***

] and shall abort attempts by a server to negotiate any TLS or SSL version prior to TLS 1.2 (RFC 5246).

FCS_TLSC_EXT.1.2 The TSF shall be able to support the following [

- ***TLS 1.2 ciphersuites: [***
  - ***CNSA 1.0 compliant [***
    - ***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 and RFC 8422***
    - ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 and RFC 8422***

  ]

,

] offering the supported ciphersuites in a ClientHello message in preference order: [*TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384*, *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384*].

FCS_TLSC_EXT.1.3 The TSF shall not offer ClientHello messages indicating the following:

- null encryption

- support for anonymous servers
- use of cryptography that is deprecated, export-grade, or otherwise disallowed for encryption, including DES, 3DES, RC2, RC4, or IDEA
- use of MD5 or SHA-1 for key derivation

and shall abort sessions where a server attempts to negotiate ciphersuites not enumerated in the ClientHello message.

FCS_TLSC_EXT.1.4 The TSF shall be able to support the following TLS ClientHello message extensions:

- signature_algorithms extension (RFC 8446) indicating support for CNSA 1.0 compliant [
  - ***ecdsa_secp384r1_sha384 (RFC 8446)***
  - ***rsa_pkcs1_sha384 (RFC 8446)***

  ], and [
  - ***CNSA 1.0 compliant [***
    - ***rsa_pss_pss_sha384 (RFC 8446)***
    - ***rsa_pss_rsae_sha384 (RFC 8446)***

    ]
    - ***[rsa_pss_pss_sha256, rsa_pss_rsae_sha512, rsa_pkcs1_sha256, rsa_pkcs1_sha512, ecdsa_secp521r1_sha512]***

  ], and

  [
  - ***supported_groups extension indicating support for [***
    - ***CNSA 1.0 compliant [***
      - ***secp384r1 (RFC 8446)***

      ]
    - ***and [***
      - ***secp521r1 (RFC 8446)***

      ]
  - ***extended_master_secret extension (RFC7627) enforcing server support, and [no other enforcement mode]***

  ] and shall not send the following extensions:
  - early_data
  - psk_key_exchange_modes indicating PSK only mode.

FCS_TLSC_EXT.1.5 The TSF shall be able to [

- ***verify that a presented identifier of name type: [***
  - ***dNSName according to RFC 6125***
  - ***IPAddress according to RFC 5280***

  ]

] matches a reference identifier for the requested TLS server and shall abort the session if no match is found.

FCS_TLSC_EXT.1.6 The TSF shall not establish a trusted channel if [

- ***the server certificate is invalid [with no TLS-specific exceptions]***

].

FCS_TLSC_EXT.2 - TLS Client Protocol with Mutual Authentication

FCS_TLSC_EXT.2.1 The TSF shall support mutual TLS authentication using X.509v3 certificates during the handshake and [***at no other time***], in accordance with [***RFC 5246, Section 7.4.4***].

FCS_TLSC_EXT.4 - TLS Client Support for Renegotiation

FCS_TLSC_EXT.4.1 The TSF shall support secure TLS renegotiation through use of [

- ***rejection of all renegotiation attempts***

] and shall terminate the session if an unexpected ServerHello is received or [***hello request message is received***].

5.2.2 User Data Protection (FDP)

FDP_DEC_EXT.1 – Access to Platform Resources

FDP_DEC_EXT.1.1 The application shall restrict its access to [

- ***network connectivity***

].

FDP_DEC_EXT.1.2 The application shall restrict its access to [

- ***no sensitive information repositories***

].

Application Note*: The TOE is not designed to be installed on a mobile device (such as a phone or tablet).*

FDP_NET_EXT.1 – Network Communications

FDP_NET_EXT.1.1 The application shall restrict network communication to [

- ***[communication with Server]***

].

FDP_DAR_EXT.1 – Encryption of Sensitive Application Data

FDP_DAR_EXT.1.1 The application shall [

- ***not store any sensitive data***

] in non-volatile memory.

5.2.3 Identification and Authentication (FIA)

FIA_X509_EXT.1– X.509 Certificate Validation

FIA_X509_EXT.1.1 The TSF shall [***implement functionality***] to validate certificates in accordance with the following rules:

- Certification path validation meets requirements of RFC 5280 for certificate paths of [***unlimited path length***] and certificate paths exceeding the maximum path length are invalid.
- The current time is within the notBefore and notAfter values of all certificates in the certification path.
- The certification path shall terminate at a trust anchor element appropriate for the supported function.
- Certificates containing subjectUniqueID and issuerUniqueID fields are considered invalid.
- Certificates are signed using cryptographic signatures and hashes in accordance with RFC 8603, and [
  - ***no other algorithms***

  ] and certificates signed using other cryptographic algorithms are considered invalid.
- [
  - ***no other algorithm constraints***

  ]
-

FIA_X509_EXT.1.2 The TSF shall [***implement***] processing of the extensions indicated in RFC 5280, section 4.2,

- Authority Key Identifier,
- Subject Key Identifier
- keyUsage

  and [
  - ***basicConstraints***
  - ***authorityInformationAccess***
  - ***Subject alternative name containing any of the following name types***
    [
    - ***dNSName***
    - ***iPAddress***

    ]
  - ***extendedKeyUsage***

  ].

FIA_X509_EXT.1.3 The TSF shall [***implement functionality***] to validate revocation status of the certificate using [

- ***The Online Certificate Status Protocol (OCSP) as specified in RFC 6960***

]

FIA_X509_EXT.1.4 The TSF shall [

- ***implement functionality***

] to obtain supported revocation status information via [

- ***Network connection to [OSCP responder]***

]

FIA_X509_EXT.1.5 The TSF shall [***implement functionality***] to validate that the context of the certificate path and trust store element is consistent with the supported function use via [

- ***processing [***
  - ***extendedKeyUsage (EKU) field constraints in the leaf certificate including: [***
    - ***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),***
    - ***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),***

  ]

] and rejects the certificate if the context requirements are not met,

].

FIA_X509_EXT.1.6 The TSF shall [***use platform-managed trust stores***] used for all certification path validation.

FIA_X509_EXT.2 – X.509 Certificate Support for Functions

FIA_X509_EXT.2.1 The TSF shall [***validate***] X.509v3 certificates in accordance with FIA_X509_EXT.1 to support [*authentication of sessions*] using [

- ***[TLS]***

]

FIA_X509_EXT.2.2 For each function indicated in FIA_X509_EXT.2.1, the TSF shall [***determine***] whether the [***administrator is allowed to configure certificate acceptance***] when valid certificate revocation status information cannot be obtained from a source indicated in FIA_X509_EXT.1.3.

FIA_X509_EXT.3 – X.509 Certificate Requests

FIA_X509_EXT.3.1 The TSF shall [***invoke the TOE platform to generate***] Certificate Requests as specified by [

- ***RFC 2986 (PKCS-10)***

] and be able to provide the following information in the request: public key, [

- ***Subject DN consisting of values for [***
  - ***O***
  - ***OU***
  - ***CN***
  - ***[State]***

  ]
- ***one or more of the following SAN types [***
  - ***dNSName***
  - ***iPAddress***
  - ***[no other types]***

  ]
- ***and [***
  - ***no other information.***

  ]

]

FIA_X509_EXT.3.2 The TSF shall **[*invoke platform-provided functionality*]** to validate the certificate path in accordance with FIA_X509_EXT.1 upon receiving the CA Certificate Response, and validate that the certificate path of the response is consistent with the intended usage.

FIA_XCU_EXT.1 – Implementation of X.509 Functions

FIA_XCU_EXT.1.1 The TSF shall [***verify***, ***assert***] identities included in X.509 certificates.

FIA_XCU_EXT.2 – X.509 Certificate Acquisition

FIA_XCU_EXT.2.1 The TSF shall [

- ***request certificates from an [external] CA***

] to represent [*itself for TLS authentication*] for [

- ***TLS***

]

5.2.4 Security Management (FMT)

FMT_MEC_EXT.1 – Supported Configuration Mechanism

FMT_MEC_EXT.1.1 The application shall [

- ***invoke the mechanisms recommended by the platform vendor for storing and setting configuration options.***

].

FMT_CFG_EXT.1 – Secure by Default Configuration

FMT_CFG_EXT.1.1 The application shall [***not use credentials***].

FMT_CFG_EXT.1.2 The application shall be configured by default with file permissions which protect the application binaries and data files from modification by normal unprivileged users.

FMT_SMF.1 – Specification of Management Functions

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

- ***[TLS/X509 configuration***
- ***Querying the current version of the TOE]***

]

5.2.5 Privacy (FPR)

FPR_ANO_EXT.1 – User Consent for Transmission of Personally Identifiable Information

FPR_ANO_EXT.1.1 The application shall [

- ***not transmit PII over a network***

].

5.2.6 Protection of the TSF (FPT)

FPT_API_EXT.1 – Use of Supported Services and APIs

FPT_API_EXT.1.1 The application shall use only documented platform APIs.

FPT_AEX_EXT.1 – Anti-Exploitation Capabilities

FPT_AEX_EXT.1.1 The application shall not request to map memory at an explicit address except for [*no exceptions*].

FPT_AEX_EXT.1.2 The application shall [

- ***not allocate any memory region with both write and execute permissions***

].

FPT_AEX_EXT.1.3 The application shall be compatible with security features provided by the platform vendor.

FPT_AEX_EXT.1.4 The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.

FPT_AEX_EXT.1.5 The application shall be compiled with stack-based buffer overflow protection enabled.

FPT_FLS.1 – Failure with Preservation of Secure State

FPT_FLS.1.1 The TSF shall preserve a secure state when the following type of failures occur: [*DRBG self-test failure*].

FPT_TST.1 – TSF Self-Testing

FPT_TST.1.1 The TSF shall run a suite of the following self-tests [***during initial start-up***] to demonstrate the correct operation of [*TSF DRBG specified in FCS_RBG.1*].

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of [*[DRBG seed/output data]*].

FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of [*[TSF DRBG specified in FCS_RBG.1]*].

FPT_TUD_EXT.1 – Support for Trusted Updates

FPT_TUD_EXT.1.1 The application shall [***use platform-provided services***] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2 The application shall [***provide the ability***] to query the current version of the application software.

FPT_TUD_EXT.1.3 The application shall [

- ***not download, modify, replace or update its own binary code***

].

FPT_TUD_EXT.1.4 Application updates shall be digitally signed such that the application platform can cryptographically verify them prior to installation.

FPT_TUD_EXT.1.5 The application is distributed [***as an additional software package to the platform OS***].

FPT_TUD_EXT.2 – Integrity for Installation and Update

FPT_TUD_EXT.2.1 The application shall be distributed using the [***format of the platform-supported package manager***].

FPT_TUD_EXT.2.2 The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuration settings, output files, and audit/log events.

FPT_TUD_EXT.2.3 The application installation package shall be digitally signed such that its platform can cryptographically verify them prior to installation.

FPT_LIB_EXT.1 – Use of Third Party Libraries

FPT_LIB_EXT.1.1 The application shall be packaged with only [*libraries identified in vendor-provided SBOM*]

5.2.7 Trusted Path/Channel (FTP)

FTP_DIT_EXT.1 – Protection of Data in Transit

FTP_DIT_EXT.1.1 The application shall [

- ***encrypt all transmitted [data] with [***
  - ***TLS as a client defined in the Functional Package for Transport Layer Security (TLS), version 2.1***

] for [protocol authentication] using certificates as defined in the Functional Package for X.509

] between itself and another trusted IT product.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are included by reference to [APPSW].

Table 2 Assurance Components

Requirement Class	Requirement Component
ASE: Security Target	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.1 Security objectives
	ASE_REQ.1 Security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ADV: Development	ADV_FSP.1 Basic functional specification
AGD: Guidance Documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-Cycle Support	ALC_CMC.1 Labelling of the TOE
	ALC_CMS.1 TOE CM coverage
	ALC_FLR.3 Systematic flaw remediation
	ALC_TSU_EXT.1 Timely security updates
ATE: Tests	ATE_IND.1 Independent testing - conformance
AVA: Vulnerability Assessment	AVA_VAN.1 Vulnerability survey

6. TOE Summary Specification

This chapter describes the security functions:

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

6.1 Cryptographic Support

FCS_CKM_EXT.1 FCS_CKM.1/AK FCS_CKM.1/SK	The TOE does not provide users with the capability to generate certificates[4]. Platform Administrators can set/load client or server certificates into the certificate store of the platform (i.e., keypair generated on the platform) that the TOE is running on. During a TLS handshake, the TOE utilizes ECC scheme to generate asymmetric keys with NIST curves that include P-384 and P-521 that adhere to the NIST Special Publication 186-5. For details regarding the algorithms supported and their CAVP certificates, see table below.
FCS_CKM.2	The TOE implements key establishment methods using elliptical curve key establishment scheme (ECDHE). The curves utilized by the TOE include P-384 and P-521 as defined in NIST SP 800-56A (revision 3).
FCS_COP.1/SKC	The TOE can encrypt/decrypt using AES-GCM mode (as defined in NIST SP 800-38D) with key sizes 256-bits. Corresponding CAVP certificates for these algorithms are present in the table below.
FCS_COP.1/Hash	The TOE uses hash functions that include SHA-256, SHA-384, and SHA-512 as defined in FIPS 180-4. The digest sizes include 256-bits, 384-bits, and 512-bits that are compliant with FIPS 180-4. The hashing capabilities are utilized for digital signature verification and generation and software integrity checks. SHA-1 is not used for generating digital signatures as noted in SP 800-131A but is used only for verification for legacy purposes. The TOE uses SHA-384 hashing as part of generating digital signatures. SHA-512 is used as part of the software integrity power-up test. Corresponding CAVP certificates for these algorithms are present in the table below.
FCS_COP.1/KeyedHash	The TOE supports the use of a Keyed-Hash Message Authentication algorithms that include HMAC-SHA-384, and HMAC-SHA-512. They include key sizes of 384-bits, and 512-bits respectively. The HMAC-SHA functions are used as part of the TOE's integrity check (HMAC-SHA-512) to ensure that it has not been tampered and is additionally used as part of the TLS handshake (HMAC-SHA-384). Corresponding CAVP certificates for these algorithms are present in the table below.

[4] The TOE does come with pre-installed self-signed certificates that it protects with a randomly generated passphrase. This passphrase can be encrypted in its database using configuration setting. This self-signed certificate is intended for first-time startup, and the customers should replace this certificate with their own.

FCS_COP.1/SigGen FCS_COP.1/SigVer	Both RSA and ECDSA schemes are used for TLS functions with Approved key sizes. These include RSA 3072-bits and 4096-bits. For ECDSA, they include the curve sizes P-384 and P-521. During TLS handshakes, these certificates are used for peer authentication to verify the server's identity (and/or client, if configured). These certificates are also used by the TOE to present its identity to the peer. Corresponding CAVP certificates and the relevant schemes for these algorithms are present in the table below.
FCS_RBG_EXT.1 FCS_RBG.1 FCS_RBG.2	The TOE implements DRBG functionality using the CTR_DRBG in AES-256 mode. The DRBG is seeded using the platform's entropy source (outside of TOE's boundary), which provides a minimum of 256 bits of entropy. The entropy design source description is described in a separate document proprietary to the platform vendor.
FCS_STO_EXT.1	The TOE does not store any credentials.
FCS_TLSC_EXT.1 FCS_TLSC_EXT.2 FCS_TLSC_EXT.4	All data that is transmitted between the TOE and Server, in the operational environment, are encrypted using TLSv1.2. When the TOE is establishing a TLS session, it checks the reference identifier that has been specified by the administrators. These reference identifiers can include FQDN or IP address (not recommended) and are checked by looking at the Subject Alternative Name (SAN). The certificate must have a SAN field, or the TLS connection will be terminated. The TOE does not support wildcards if a certificate is presented with one in it. Certificate pinning is also not supported and session renegotiation is not allowed . The extended_master_secret extension is also supported. The TOE does not establish a trusted channel if the server certificate is invalid – no exceptions. CA certificates are stored in the Linux trust anchor or key store. During the TLS handshake, the TOE presents the following four cipher suites in its Client Hello. For the cipher suites noted below, the client hello extension supports secp384r1 and secp521r1 NIST curves and signature_algorithms identified in the SFR above. • *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384* • *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384* During the negotiation of the TLS handshake, X.509v3 certificates are used to verify the server's identity. Additionally, client certificates should be configured on the TOE platform to support mutual authentication.

Table 3 Cryptographic Functions and CAVP Certificates

Algorithm Name and Capabilities	Algorithm Description	Certificate
Asymmetric key generation (FCS_CKM.1 and FCS_CKM.1/AK)		
ECDSA KeyGen and KeyVer (P-384, P-521 curves)	FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2	#A8208
Cryptographic key establishment (FCS_CKM.2)		

Elliptic Curve Diffie-Hellman with NIST Curves P-384 and P-521	NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	#A8208
Symmetric encryption/decryption (FCS_COP.1/SKC)		
AES-GCM (256 bits)	Advanced Encryption Standard (FIPS PUB 197) GCM as defined in NIST SP 800-38D	#A8208
Cryptographic hashing (FCS_COP.1/Hash)		
SHA-384 SHA-512	Secure Hashing Algorithm (FIPS PUB 180-4)	#A8208
Cryptographic signature services (FCS_COP.1/Sig)		
RSA SigGen and SigVer with 3072-bit modulus	Rivest–Shamir–Adleman (FIPS PUB 186-5)	#A8208
ECDSA SigGen and SigVer with NIST Curves P-384 and P-521	Elliptical Curve Digital Signature Algorithm (FIPS PUB 186-5)	#A8208
Keyed-hash message authentication (FCS_COP.1/KeyedHash)		
HMAC-SHA-384 HMAC-SHA-512	Hash-based Message Authentication Code FIPS PUB 198-1 FIPS PUB 180-4	#A8208
Deterministic random bit generation (FCS_RBG.1)		
Counter-DRBG (AES-256)	Deterministic Random Bit Generator NIST SP 800-90A	#A8208

6.2 User Data Protection

FDP_DEC_EXT.1	The TOE accesses the network connectivity of its platform as identified by the SFR. The TOE does not access any sensitive information repositories.
FDP_NET_EXT.1	The TOE can respond to TLS connections with Server in the operational environment.
FDP_DAR_EXT.1	The TOE does not store sensitive data.

6.3 Identification and Authentication

FIA_X509_EXT.1 FIA_X509_EXT.2 FIA_X509_EXT.3	The TOE implements the ability to perform certificate path validation on the certificate chain that is presented to it by the peer. The certificate path validation begins with the identity certificate presented by the peer, and then proceeds in checking the intermediate CA certificate(s) until it reaches the trusted CA certificate stored in the platform OS trust store. If no trusted CA certificate is found (i.e., the

	TOE cannot choose to use any certificate for path validation), the session will terminate. The following steps are performed for each certificate in the path: • The public key algorithm/parameters are checked (i.e., RSA/ECDSA key sizes meet requirements of 2048 bits or greater for RSA and 256 bits or greater for ECDSA) • The certificate is checked to make sure it is not expired (i.e., validity period of the certificate must be proper) • The certificate is checked to make sure it is not revoked using OCSP • The issuer name is checked to ensure that it matches the subject name of the previous certificate in the chain • The certificate is checked that it terminates with a trusted CA certificate and that all CA certificate have the basicConstraints extension present (and set to TRUE) • The extendedKeyUsage field is checked such that OCSP certificates and server/client certificates contain the correct OID (e.g., OCSP Signing purpose and Server/Client Authentication purpose) • The key usage extension of the certificate is checked to make sure that it is allowed to sign certificates Certificates that are presented to the TOE must meet the x509v3 requirements as defined in RFC 5280 for TLS. If there are any issues with the certificate presented (as noted above), the TOE will not accept the certificate and reject the connection. A log message will be generated, and an administrator will be required to address the problem noted for the connection to succeed. The TOE also supports revocation checking of the certificate presented using OCSP as specified in RFC 2560. In the event that the certificate is revoked following a check of its status, the TOE will terminate the connection. If the OCSP responder can't be reached, the administrator can configure if the connection is established or terminated prior to the event occurring. As the TOE is a software application, it depends on the platform to generate the CSR (Certificate Signing Request) using PKCS-10 format and path validation consistent with intended usage. The CSR may include Organization name (O), Department (OU), State (S), but Common Name (CN), Subject Alternative Name (SAN), and public key are required. In addition, the SAN types DNS and/or IP addresses are supported. For the purpose of the functions, only these two SAN types are applicable in the evaluated configuration. Peer certificates must contain SAN or the TLS session will not be established.
FIA_XCU_EXT.1 FIA_XCU_EXT.2	The TOE will verify the peer certification for TLS authentication. In addition, the TOE supports mutual authentication and will assert its own identity as well. For its own X.509 certificate, the TOE requests a certificate from an external CA via the platform.

6.4 Security Management

FMT_MEC_EXT.1	The TOE stores its configuration settings and logs in Linux recommended directories.
FMT_CFG_EXT.1	The TOE does not use credentials. The TOE relies on the operating system to enforce I&A and provide configurations for privileged users. By default, the TOE includes file permissions that protect the TOE's binary and data files from modification from normal unprivileged users. The TOE also includes an integrity check for itself to ensure that no malicious activity occurs.
FMT_SMF.1	The TOE provides several management functions that include the following: ▪ TLS/X509 configuration ▪ Querying the current version of the TOE

6.5 Privacy

FPR_ANO_EXT.1	The TOE does not transmit or store any personally identifiable information about an individual.

6.6 Protection of the TSF

FPT_API_EXT.1	The TOE includes the use of documented Linux system APIs (kernel version 2.6.26 or later). System function calls includes functionality such as • Driver, • Data type, • C library functions, • Kernel functions, • Memory management, • Kernel IPC, • Hardware, and • Firmware.
FPT_AEX_EXT.1	The TOE automatically enables ASLR when the application is compiled (with -pie build flag), and stack-based buffer overflow protection is enabled via Golang language. The TOE does not request any memory mapping at an explicit address. The TOE does not allocate any memory region with both write and execute permissions; users shall also not write user-modifiable files to directories that contain executable files unless they are explicitly told to do so. The TOE is designed to be compatible with the security features that are provided by the platform (SELinux or AppArmor) vendor that is it installed on.
FPT_FLS.1 FPT_TST.1	The DRBG self-test and failure detection and description are described in a separate document proprietary to the platform vendor. If any of the self-tests fails, the TOE

	will not start up (i.e., in FIPS error state). Please reboot the system or re-install the TOE if the failure is persistent.
FPT_TUD_EXT.1 FPT_TUD_EXT.2	The TOE has specific versions, which can be queried by the user via the TOE's interface. New versions of the TOE are created by Palo Alto Networks, which an administrator can retrieve to update the current version of the TOE. During the installation process, a digital signature verification check can be performed by the platform to verify that the signed installer has not been modified. All versions of the installer are digitally signed by Palo Alto Networks . Customer Support will send verified customers an email with a download link. They also send license key and instructions in the email. The link is only valid for a limited number of downloads and time. The image is securely retrieved from the download server (https://download.demisto.com) The TOE cannot update its own binary code – it relies on the administrator to download and install the new version available. Registered users are emailed when a security update has been published and available on the download server. Customers with support account in the Customer Support Portal (https://support.paloaltonetworks.com/) will also get notifications as well. Based on the Linux distribution, the package can be RPM (e.g., Redhat) or DEB (e.g., Ubuntu).
FPT_LIB_EXT.1	The TOE utilizes Golang (version 1.25) for its TLS functions and Go Crypto module (version 1.0.0 [5]) for its crypto functions.

[5] https://csrc.nist.gov/projects/cryptographic-module-validation-program/certificate/5247

6.7 Trusted Path/Channel

FTP_DIT_EXT.1	All data transmitted between the TOE and Server, in the environment, are encrypted using TLSv1.2. For more details, please see the FCS_TLS* requirements above.

7. Protection Profile Claims

This ST is conformant to the [APPSW].

8. Rationale

This Security Target includes by reference the [APPSW] Security Problem Definition, Security Objectives, and Security Assurance Requirements. The security target makes no additions to the [APPSW] assumptions. Security functional requirements have been reproduced verbatim with the protection profile operations completed. Operations on the security requirements follow [APPSW] application notes and assurance activities. The security target did not add or remove any security requirements. Consequently, [APPSW] rationale applies and is complete.