Axway Validation Authority Server, version 5.2 Security Target

Version 0.4
07/02/2024

Prepared for:

Axway, Inc.

16220 N Scottsdale Road, Ste 500
Scottsdale, AZ 85254

Prepared By:

www.gossamersec.com

LIST OF TABLES

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 Axway Validation Authority Server, version 5.2 provided by Axway, Inc.. The TOE is being evaluated as a software applications.

The Security Target contains the following additional sections:

- Conformance Claims (Section 2)
- Security Objectives (Section 3)
- Extended Components Definition (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

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 parenthetical number 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.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold 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 italics 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 …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.1 Security Target Reference

ST Title – Axway Validation Authority Server, version 5.2 Security Target

ST Version – Version 0.4

ST Date – 07/02/2024

1.2 TOE Reference

TOE Identification –Axway Validation Authority Server, version 5.2

TOE Developer – Axway, Inc.

Evaluation Sponsor – Axway, Inc.

1.3 TOE Overview

The Target of Evaluation (TOE) is the Axway Validation Authority Server, version 5.2.

1.4 TOE Description

The Axway Validation Authority Server (Server) is part of Axway's Validation Authority Suite, which provides a comprehensive, scalable, and reliable framework for real-time validation of digital certifications for the Public Key Infrastructure (PKI). The Axway VA Suite provides a variety of PKI and certificate management functionality to prevent revoked credentials from being used for secure email, smart card login, network access (including wireless), or other sensitive electronic transactions. The administrator can configure the Axway VA Server to act in one of two manners: Repeater or Responder. One can think of the Repeater, conceptually the simpler configuration, as a revocation caching proxy (locally caching CRLs and OCSP responses). While the Responder can locally cache CRLs and generate new OCSP responses (using the certificate statuses within the CRLs) for clients.

The TOE provides the following functionality:

- Maintains and processes a store of digital certificate revocation data by obtaining the digital Certificate Revocation List (CRL) from multiple CA or VA sources and performing end-to-end certificate validation if one or more intermediate CAs are used and the validation policy requires a complete certificate chain validation.
- Generates and signs OCSP/SCVP responses. Maintains a cache loaded with OCSP responses that are pre-computed or dynamically built up by proxy client requests to a responder.
- Allows caching of CRLs and delta CRLs to support non-OCSP clients or clients that want to maintain their own revocation data caches for backup and in low-bandwidth and non real-time environments.
- Supports SSL-based communications with clients, digitally signed client requests/responses, and digitally signed XML logs and CRL archives, as well as SSL-based server administration.
- Supports software PKCS #11 or CAPI token based hardware signing and encryption products, including hardware security modules from leading vendors that comply with FIPS 140-2 Level 2 or above.[1]

For purposes of this evaluation, the Axway Server is a software application that offers cryptographic functions (key generation, hashing, signing, random bit generation), secure remote administration, secure storage of credentials, X.509 certificate validation and authentication, trusted update, anti-exploitation capabilities and restricted network communications. This evaluation is limited to the security functions claimed in Section 5 and further described in Section 6 of this Security Target (ST).

1.4.1 TOE Architecture

The Axway Validation Authority Server runs on the following platforms:

- Microsoft Windows Server 2019 (64 bit) on VMWare ESXi 7.0 running on Intel Xeon E5-2670
- Microsoft Windows Server 2022 (64 bit) on VMWare ESXi 7.0 running on Intel Xeon E5-2670
- RHEL 7 (64 bit) on VMWare ESXi 7.0 running on Intel Xeon E5-2670
- RHEL 8 (64 bit) on VMWare ESXi 7.0 running on Intel Xeon E5-2670
- RHEL 9 (64 bit) on VMWare ESXi 7.0 running on Intel Xeon E5-2670

The Windows and RHEL platforms are part of the operating environment of the TOE. The TOE can execute on any Intel Xeon processor, however the lab tested the TOE on an Intel Xeon E5-2670. The lab also tested the TOE on Windows Server 2022 (64 bit) and RHEL 8 (64 bit) running on VMWare ESXi 7.0 in the evaluated configuration.

[1] The use of a Hardware Security Module (HSM) is not included in the evaluated configuration.

The TOE binaries remain unchanged for each flavor of operating systems. Thus, the same TOE binaries compiled as Windows executables are used for all claimed Windows operating systems, and the same TOE binaries compiled as 64-bit ELF executables are used for all claimed Red Hat Linux distributions.

The Axway VA Suite is composed of the following applications:

1. Validation Authority Server (VA Server) – the VA Server is comprised of the VA validation server acting as either a Repeater or Responder operating on a Windows or Linux platform, and the Web based administration (Admin UI). The VA Server maintains a store of digital certificate revocation data and ensures the integrity and validity of online transactions by delivering real-time validation of digital certificates.
2. Desktop Validator (DV) - (Standard and Enterprise Editions) - the Desktop Validator is a Microsoft CAPI compliant revocation trust provider that communicates with the Validation Authority Server (VA server) in responder mode to check status of digital certs in real time. DV runs as a service on a 64bit Microsoft Windows platforms and can be invoked to validate standard X.509v3 digital certificates issued by any Certificate Authority (CA). The DV Standard edition provides certificate validation support for client applications, while the DV Enterprise edition provides certificate validation support for both client and server applications.

The focus of the evaluation is the Validation Authority Server (VA Server). The diagram below shows the TOE's interaction with components in its environment.

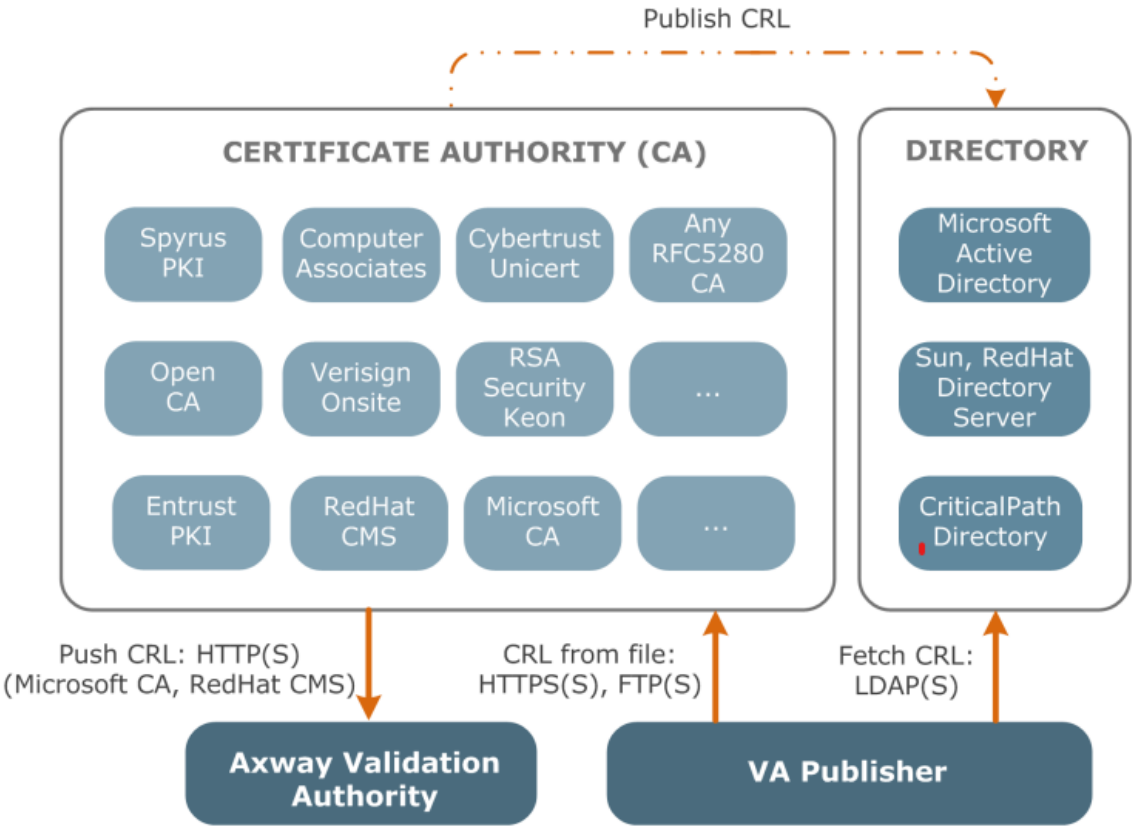

Figure 1 - Axway Validation Authority Server (VA Server) Diagram

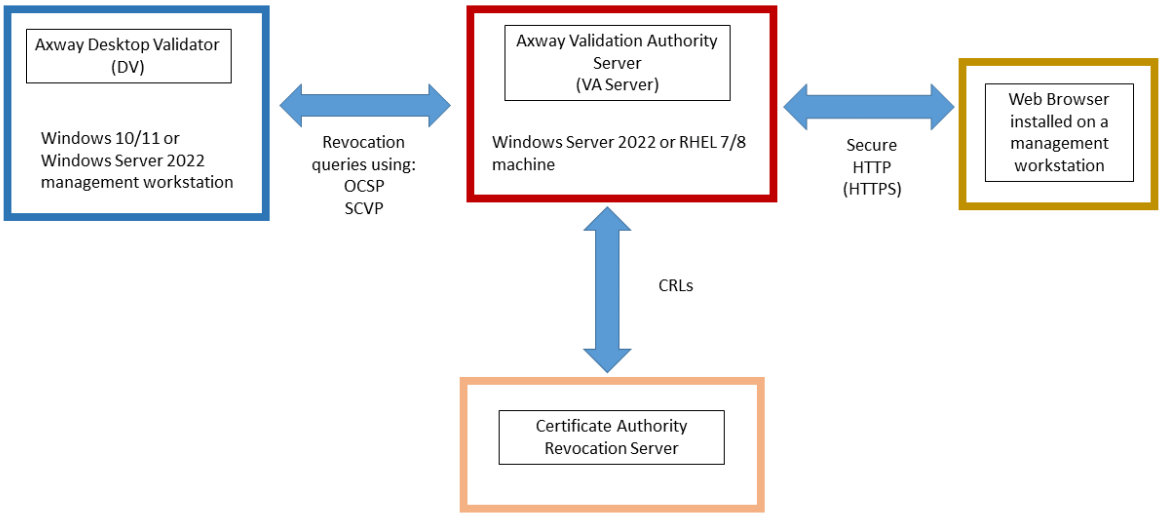

Figure 2 - Axway Validation Authority (VA) Server interaction with additional components

The cryptographic capabilities of Axway VA Server are provided by the Axway OpenSSL version 3.0.13 (with 3.0.8 FIPS), which is a software cryptographic module that is implemented as two dynamic link libraries (DLLs) on Windows or two Shared Objects (SOs) on Linux. It is a user space shared library built from OpenSSL 3.0. The implementation of TLS/HTTPS to secure communication channels is supported using Apache.
The environmental components described in the following table are required to operate the TOE in the evaluated configuration.

Component	Description
Axway Desktop Validator (DV) Client (Mandatory)	The Axway Desktop Validator Client is another application in the Axway Validation Authority Suite. The DV interfaces with the TOE to use the VA server's certificates for outgoing revocation queries.
Certificate Authority Revocation Server (Mandatory)	Validity of the certificates the TOE uses for asserting the authenticity of the TLS peers is verified using CRLs.
Management Workstation (Mandatory)	A workstation used by an administrator to locally or remotely manage the TOE. The workstation must have a compatible web browser to connect to the TOE's web UI for management.

Table 1 IT Environment Components

1.4.1.1 Physical Boundaries

The TOE is a software-only application which executes on a Microsoft Windows or RHEL operating system platform. The underlying platform is considered part of the operating environment but provides some of the security functionality required by the ASPP14. The evaluated configuration includes the Axway Validation Authority Server v5.2 - a software server application running on the platforms listed above in the TOE Architecture section.

The TOE also requires a Certificate Authority (CA) Revocation server in the operational environment to provide provide the revocation status of valid digital certificates.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by the Axway Validation Authority Server:

- Cryptographic support
- User data protection

- Identification and authentication
- Security management
- Privacy
- Protection of the TSF
- Trusted path/channels

1.4.1.2.1 Cryptographic support

The TOE uses CAVP-validated cryptographic algorithm implementations, provided the Axway OpenSSL version 3.0.13 (with 3.0.8 FIPS), to support asymmetric key generation, encryption/decryption, signature generation and verification and establishment of trusted channels to protect data in transit. The TOE provides a web server for TLS/HTTPS to facilitate trusted remote communications and implements functionality to securely store key data related to secure communications. The TOE also relies on the underlying platform to generate entropy that is used as input data for the TOE's deterministic random bit generator (DRBG).

1.4.1.2.2 User data protection

The TOE does not access any hardware resources (other than network connectivity) or sensitive information repositories. The TOE does not store any sensitive data in non-volatile memory. Inbound and outbound network communications are restricted to those that are application initiated.

1.4.1.2.3 Identification and authentication

The TOE implements X509 certificate validation to validate the revocation status of certificates using CRL. The TOE uses X509 certificates to support HTTPS/TLS authentication of administrators.

1.4.1.2.4 Security management

The TOE provides a Web-based Graphical User Interface (Web GUI) to access and manage the TOE security functions. When configured with default credentials or no credentials, the TOE restricts its functionality and only allows the ability to set new credentials. By default, the TOE is configured with file permissions to protect itself and its data from unauthorized access.

1.4.1.2.5 Privacy

The TOE does not transmit personally identifiable information (PII) over any network interfaces.

1.4.1.2.6 Protection of the TSF

The TOE protects itself against exploitation by implementing address space layout randomization (ASLR) and by not allocating any memory region for both write and execute permission. The TOE is compiled for both Windows and Linux with stack-based buffer overflow protection and does not allow user-modifiable files to be written to directories that contain executable files. The TOE uses standard platform APIs and includes a number of third party libraries used to perform its functions.

The TOE includes mechanisms to check for updates and to query the current version of the application software. TOE software is digitally signed and distributed using the platform-supported package manager (Windows or Linux). The TOE does not update its own binary code in any way and when removed, all traces of the TOE application software are deleted.

1.4.1.2.7 Trusted path/channels

The TOE protects communications between itself and remote administrators using HTTPS/TLS.

1.4.2 TOE Documentation

The following user and administrative guidance is available:

- Validation Authority Version 5.2 Common Criteria Guide, July 1, 2024

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
    - Part 3 Extended
- Package Claims:
    - 'Protection Profile for Application Software', Version 1.4, 07 October 2021/'Functional Package for Transport Layer Security (TLS)', Version 1.1, 12 February 2019 (ASPP14/PKGTLS11)

Package	Technical Decision	Applied	Notes
PP_APP_V1.4	TD0823	Yes	Updated link in FPT_AEX_EXT.1.3 test description
PP_APP_V1.4	TD0822	Yes	Updated Windows manifest file name in tests FDP_DEC_EXT.1.1 and FDP_DEC_EXT.1.2
PP_APP_V1.4	TD0815	Yes	Added a TSS activity and modified the test for FPT_AEX_EXT.1.5
PP_APP_V1.4	TD0798	Yes	Applies to evaluation activities only
PP_APP_V1.4	TD0780	Yes	Applies to FIA_X509_EXT.1 test 4 only
PP_APP_V1.4	TD0756	Yes	Applies to test evaluation activity only
PP_APP_V1.4	TD0747	Yes	Applies to test description for Android platform only
PP_APP_V1.4	TD0743	Yes	Changes to FTP_DIT_EXT.1 selections and Application Note
PP_APP_V1.4	TD0736	Yes	Addition of FCS_HTTPS_EXT.1.3/Server
PP_APP_V1.4	TD0719	No	PP document updated to include Extended Component Definitions appendix
PP_APP_V1.4	TD0717	Yes	Formatting changes to FCS_CKM.1 and FCS_COP SFRs
PP_APP_V1.4	TD0664	Yes	Applies to tests only
PP_APP_V1.4	TD0650	Yes	New module claims
PP_ASPP_V1.4	TD0628	Yes	Applies to tests only
PKG_TLS_V1.1	TD0779	Yes	Adjustments in selections
PKG_TLS_V1.1	TD0770	Yes	Adjustment to FCS_TLSS_EXT.2.2 SFR app note and evaluation activities
PKG_TLS_V1.1	TD0739	Yes	Change to FCS_TLSS_EXT.1.3 test 1
PKG_TLS_V1.1	TD0726	Yes	Updates to selections in FCS_TLSS_EXT.1.3
PKG_TLS_V1.1	TD0513	No	TLSC not claimed
PKG_TLS_V1.1	TD0499	No	FCS_TLSC_EXT.1 not claimed
PKG_TLS_V1.1	TD0469	Yes	Test removed
PKG_TLS_V1.1	TD0442	Yes	Updated TLSS ciphersuites

2.1 Conformance Rationale

The ST conforms to the ASPP14/PKGTLS11. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the PP.

3. Security Objectives

The Security Problem Definition may be found in the ASPP14/PKGTLS11 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The ASPP14/PKGTLS11 offers additional information about the identified security objectives, but that has not been reproduced here and the ASPP14/PKGTLS11 should be consulted if there is interest in that material.

In general, the ASPP14/PKGTLS11 has defined Security Objectives appropriate for software applications and as such are applicable to the Axway Validation Authority Server TOE.

3.1 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_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.

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.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the ASPP14/PKGTLS11. The ASPP14/PKGTLS11 defines the following extended requirements and since they are not redefined in this ST the ASPP14/PKGTLS11 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- ASPP14:FCS_CKM_EXT.1: Cryptographic Key Generation Services
- ASPP14:FCS_CKM_EXT.1/PBKDF: Password Conditioning
- ASPP14:FCS_HTTPS_EXT.1/Server: HTTPS Protocol
- ASPP14:FCS_HTTPS_EXT.2: HTTPS Protocol with Mutual Authentication
- ASPP14:FCS_RBG_EXT.1: Random Bit Generation Services
- ASPP14:FCS_RBG_EXT.2: Random Bit Generation from Application
- ASPP14:FCS_STO_EXT.1: Storage of Credentials
- PKGTLS11:FCS_TLS_EXT.1: TLS Protocol
- PKGTLS11:FCS_TLSS_EXT.1: TLS Server Protocol
- PKGTLS11:FCS_TLSS_EXT.2: TLS Server Support for Mutual Authentication
- ASPP14:FDP_DAR_EXT.1: Encryption Of Sensitive Application Data
- ASPP14:FDP_DEC_EXT.1: Access to Platform Resources
- ASPP14:FDP_NET_EXT.1: Network Communications
- ASPP14:FIA_X509_EXT.1: X.509 Certificate Validation
- ASPP14:FIA_X509_EXT.2: X.509 Certificate Authentication
- ASPP14:FMT_CFG_EXT.1: Secure by Default Configuration
- ASPP14:FMT_MEC_EXT.1: Supported Configuration Mechanism - per TD0747
- ASPP14:FPR_ANO_EXT.1: User Consent for Transmission of Personally Identifiable
- ASPP14:FPT_AEX_EXT.1: Anti-Exploitation Capabilities
- ASPP14:FPT_API_EXT.1: Use of Supported Services and APIs
- ASPP14:FPT_IDV_EXT.1: Software Identification and Versions
- ASPP14:FPT_LIB_EXT.1: Use of Third Party Libraries
- ASPP14:FPT_TUD_EXT.1: Integrity for Installation and Update
- ASPP14:FPT_TUD_EXT.2: Integrity for Installation and Update - per TD0664
- ASPP14:FTP_DIT_EXT.1: Protection of Data in Transit - per TD0743

Extended SARs:

- ALC_TSU_EXT.1: Timely Security Updates

5. 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 ASPP14/PKGTLS11. The refinements and operations already performed in the ASPP14/PKGTLS11 are not identified (e.g., highlighted) here, rather the requirements have been copied from the ASPP14/PKGTLS11 and any residual operations have been completed herein. Of particular note, the ASPP14/PKGTLS11 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the ASPP14/PKGTLS11. The ASPP14/PKGTLS11 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Axway Validation Authority Server TOE.

Requirement Class	Requirement Component
FCS: Cryptographic support	ASPP14:FCS_CKM_EXT.1: Cryptographic Key Generation Services
	ASPP14:FCS_CKM.1/AK: Cryptographic Asymmetric Key Generation - per TD0717
	ASPP14:FCS_CKM_EXT.1/PBKDF: Password Conditioning
	ASPP14:FCS_CKM.2: Cryptographic Key Establishment
	ASPP14:FCS_COP.1/Hash: Cryptographic Operation - Hashing – per TD0717
	ASPP14:FCS_COP.1/KeyedHash: Cryptographic Operation - Keyed-Hash Message Authentication – per TD0717
	ASPP14:FCS_COP.1/Sig: Cryptographic Operation - Signing – per TD0717
	ASPP14:FCS_COP.1/SKC: Cryptographic Operation - Encryption/Decryption – per TD0717
	ASPP14:FCS_HTTPS_EXT.1/Server: HTTPS Protocol
	ASPP14:FCS_HTTPS_EXT.2: HTTPS Protocol with Mutual Authentication
	ASPP14:FCS_RBG_EXT.1: Random Bit Generation Services
	ASPP14:FCS_RBG_EXT.2: Random Bit Generation from Application
	ASPP14:FCS_STO_EXT.1: Storage of Credentials
	PKGTLS11:FCS_TLS_EXT.1: TLS Protocol
	PKGTLS11:FCS_TLSS_EXT.1: TLS Server Protocol
	PKGTLS11:FCS_TLSS_EXT.2: TLS Server Support for Mutual Authentication
FDP: User data protection	ASPP14:FDP_DAR_EXT.1: Encryption Of Sensitive Application Data
	ASPP14:FDP_DEC_EXT.1: Access to Platform Resources
	ASPP14:FDP_NET_EXT.1: Network Communications
FIA: Identification and authentication	ASPP14:FIA_X509_EXT.1: X.509 Certificate Validation
	ASPP14:FIA_X509_EXT.2: X.509 Certificate Authentication
FMT: Security management	ASPP14:FMT_CFG_EXT.1: Secure by Default Configuration
	ASPP14:FMT_MEC_EXT.1: Supported Configuration Mechanism - per TD0747

	ASPP14:FMT_SMF.1: Specification of Management Functions
FPR: Privacy	ASPP14:FPR_ANO_EXT.1: User Consent for Transmission of Personally Identifiable
FPT: Protection of the TSF	ASPP14:FPT_AEX_EXT.1: Anti-Exploitation Capabilities
	ASPP14:FPT_API_EXT.1: Use of Supported Services and APIs
	ASPP14:FPT_IDV_EXT.1: Software Identification and Versions
	ASPP14:FPT_LIB_EXT.1: Use of Third Party Libraries
	ASPP14:FPT_TUD_EXT.1: Integrity for Installation and Update
	ASPP14:FPT_TUD_EXT.2: Integrity for Installation and Update - per TD0664
FTP: Trusted path/channels	ASPP14:FTP_DIT_EXT.1: Protection of Data in Transit - per TD0743

Table 2 TOE Security Functional Components

5.1.1 Cryptographic support (FCS)

5.1.1.1 Cryptographic Key Generation Services – per TD0717 (ASPP14:FCS_CKM_EXT.1)

ASPP14:FCS_CKM_EXT.1.1

The application shall [***implement asymmetric key generation***].

5.1.1.2 Cryptographic Asymmetric Key Generation - per TD0717 (ASPP14:FCS_CKM.1/AK)

ASPP14:FCS_CKM.1/AK.1

The **application** shall [***implement functionality***] to 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-384 and [P-256, P-521]] that meet the following: [FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4]].

5.1.1.3 Password Conditioning – per TD0717 (ASPP14:FCS_CKM_EXT.1/PBKDF)

ASPP14:FCS_CKM_EXT.1/PBKDF.1

A password/passphrase shall perform [**PBKDFv2**] in accordance with a specified cryptographic algorithm as specified in FCS_COP.1/KeyedHash, with [**2048**] iterations, and output cryptographic key sizes [***256***] that meet the following: NIST SP 800-132.

ASPP14:FCS_CKM_EXT.1/PBKDF.2

The TSF shall generate all salts using a RBG that meets FCS_RBG_EXT.1 and with entropy corresponding to the security strength selected for PBKDF in FCS_CKM_EXT.1.1/PBKDF.

5.1.1.4 Cryptographic Key Establishment (ASPP14:FCS_CKM.2)

ASPP14:FCS_CKM.2.1

The application shall [***implement functionality***] to 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 8017, 'Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1',
[Elliptic 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'].

5.1.1.5 Cryptographic Operation – Hashing – Per TD0717 (ASPP14:FCS_COP.1/Hash)

ASPP14:FCS_COP.1/Hash.1

The **application** shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384, SHA-512***] and **message digest** sizes [***160, 256, 384, 512***] bits that meet the following: [FIPS Pub 180-4].

5.1.1.6 Cryptographic Operation - Keyed-Hash Message Authentication - per TD0717 (ASPP14:FCS_COP.1/KeyedHash)

ASPP14:FCS_COP.1/KeyedHash.1

The **application** shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm [***HMAC-SHA-256, HMAC-SHA-384***] and [***HMAC-SHA-1***] **with** key sizes [**256, 384, and 160**] **and message digest sizes** [***256, 384***] **and** [***160***] bits that meet the following: [*FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4, 'Secure Hash Standard'*].

5.1.1.7 Cryptographic Operation – Signing – Per TD0717 (ASPP14:FCS_COP.1/Sig)

ASPP14:FCS_COP.1/Sig.1

The **application** shall perform [*cryptographic signature services (generation and verification)*] in accordance with a specified cryptographic algorithm [***RSA schemes using cryptographic key sizes of [2048-bit or greater] that meet the following: [FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5], ECDSA schemes using ['NIST curves' P-256, P-384 and [P-521]] that meet the following: [FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 6]***].

5.1.1.8 Cryptographic Operation - Encryption/Decryption – Per TD0717 (ASPP14:FCS_COP.1/SKC)

ASPP14:FCS_COP.1/SKC.1

The **application** shall perform [*encryption/decryption*] in accordance with a specified cryptographic algorithm
[***AES-CBC (as defined in NIST SP 800-38A) mode,***
AES-GCM (as defined in NIST SP 800-38D) mode]
and cryptographic key sizes [***128-bit, 256-bit***].

5.1.1.9 HTTPS Protocol – Per TD0736 (ASPP14:FCS_HTTPS_EXT.1/Server)

ASPP14:FCS_HTTPS_EXT.1.1/Server

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

ASPP14:FCS_HTTPS_EXT.1.2/Server

The application shall implement HTTPS using TLS as defined in the Functional Package for TLS.

ASPP14:FCS_HTTPS_EXT.1.3/Server

The application shall [***not establish the connection***] if the peer certificate is deemed invalid.

5.1.1.10 HTTPS Protocol with Mutual Authentication (ASPP14:FCS_HTTPS_EXT.2)

ASPP14:FCS_HTTPS_EXT.2.1

The application shall [***not establish the connection***] if the peer certificate is deemed invalid.

5.1.1.11 Random Bit Generation Services (ASPP14:FCS_RBG_EXT.1)

ASPP14:FCS_RBG_EXT.1.1

The application shall [***implement DRBG functionality***] for its cryptographic operations.

5.1.1.12 Random Bit Generation from Application (ASPP14:FCS_RBG_EXT.2)

ASPP14:FCS_RBG_EXT.2.1

The application shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [***CTR_DRBG (AES)***].

ASPP14:FCS_RBG_EXT.2.2

The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [***a software-based noise source***] with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

5.1.1.13 Storage of Credentials (ASPP14:FCS_STO_EXT.1)

ASPP14:FCS_STO_EXT.1.1

The application shall [***invoke the functionality provided by the platform to securely store [server password], implement functionality to securely store [HTTPS certificate's private key and OCSP response signing key] according to [FCS_COP.1/SKC]***]

to non-volatile memory.

5.1.1.14 TLS Protocol (PKGTLS11:FCS_TLS_EXT.1)

PKGTLS11:FCS_TLS_EXT.1.1

The product shall implement [***TLS as a server,***]

5.1.1.15 TLS Server Protocol (PKGTLS11:FCS_TLSS_EXT.1)

PKGTLS11:FCS_TLSS_EXT.1.1

The product shall implement TLS 1.2 (RFC 5246) and [***no earlier TLS versions***] as a server that supports the cipher suites [***TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,***
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_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
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_128_CBC_SHA256 as defined in RFC 5289,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289] and also supports functionality for [***session resumption based on session tickets according to RFC 5077***] and [***mutual authentication***] (TD0779 applied)

PKGTLS11:FCS_TLSS_EXT.1.2

The product shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [**TLS 1.1**]

PKGTLS11:FCS_TLSS_EXT.1.3

The product shall perform key establishment for TLS using [***RSA with size [2048 bits, 3072 bits, 4096 bits] and no other sizes, ECDHE parameters using elliptic curves [secp256r1, secp384r1, secp521r1] and no other curves***] (TD0726 applied)

5.1.1.16 TLS Server Support for Mutual Authentication (PKGTLS11:FCS_TLSS_EXT.2)

PKGTLS11:FCS_TLSS_EXT.2.1

The product shall support authentication of TLS clients using X.509v3 certificates.

PKGTLS11:FCS_TLSS_EXT.2.2
The product shall ***[not establish a trusted channel]*** if the client certificate is invalid. (TD0770 applied)

PKGTLS11:FCS_TLSS_EXT.2.3
The product shall not establish a trusted channel if the Distinguished Name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match one of the expected identifiers for the client.

5.1.2 User data protection (FDP)

5.1.2.1 Encryption Of Sensitive Application Data (ASPP14:FDP_DAR_EXT.1)

ASPP14:FDP_DAR_EXT.1.1
The application shall [***protect sensitive data in accordance with FCS_STO_EXT.1***] in non-volatile memory.

5.1.2.2 Access to Platform Resources (ASPP14:FDP_DEC_EXT.1)

ASPP14:FDP_DEC_EXT.1.1
The application shall restrict its access to [***network connectivity***].

ASPP14:FDP_DEC_EXT.1.2
The application shall restrict its access to [***no sensitive information repositories***].

5.1.2.3 Network Communications (ASPP14:FDP_NET_EXT.1)

ASPP14:FDP_NET_EXT.1.1
The application shall restrict network communication to [***user-initiated communication for [importing CRLs, checking for software updates], respond to [remotely administrative sessions and revocation queries]***].

5.1.3 Identification and authentication (FIA)

5.1.3.1 X.509 Certificate Validation (ASPP14:FIA_X509_EXT.1)

ASPP14:FIA_X509_EXT.1.1
The application shall [***implement functionality***] to validate certificates in accordance with the following rules:
- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The application shall validate that any CA certificate includes caSigning purpose in the key usage field
- The application shall validate the revocation status of the certificate using [***CRL as specified in RFC 5280 Section 6.3, CRL as specified in RFC 8603***]
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:
  - o 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.
  - o 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 EKU field.
  - o 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 EKU field.

o S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
o 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 EKU field.
o Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kpcmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

ASPP14:FIA_X509_EXT.1.2
The application shall treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.

5.1.3.2 X.509 Certificate Authentication (ASPP14:FIA_X509_EXT.2)

ASPP14:FIA_X509_EXT.2.1
The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [***HTTPS, TLS***].

ASPP14:FIA_X509_EXT.2.2
When the application cannot establish a connection to determine the validity of a certificate, the application shall [***accept the certificate***].

5.1.4 Security management (FMT)

5.1.4.1 Secure by Default Configuration (ASPP14:FMT_CFG_EXT.1)

ASPP14:FMT_CFG_EXT.1.1
The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

ASPP14:FMT_CFG_EXT.1.2
The application shall be configured by default with file permissions which protect the application's binaries and data files from modification by normal unprivileged user.

5.1.4.2 Supported Configuration Mechanism - per TD0747 (ASPP14:FMT_MEC_EXT.1)

ASPP14:FMT_MEC_EXT.1.1
The application shall [***invoke the mechanisms recommended by the platform vendor for storing and setting configuration options***].

5.1.4.3 Specification of Management Functions (ASPP14:FMT_SMF.1)

ASPP14:FMT_SMF.1.1
The TSF shall be capable of performing the following management functions [***Configure Users, Configure Password Policy, Configure Certificates, Configure revocation sources, Create/Import Key pair, Check for Software Updates***].

5.1.5 Privacy (FPR)

5.1.5.1 User Consent for Transmission of Personally Identifiable (ASPP14:FPR_ANO_EXT.1)

ASPP14:FPR_ANO_EXT.1.1
The application shall [***not transmit PII over a network***].

5.1.6 Protection of the TSF (FPT)

5.1.6.1 Anti-Exploitation Capabilities (ASPP14:FPT_AEX_EXT.1)

ASPP14:FPT_AEX_EXT.1.1
The application shall not request to map memory at an explicit address except for [**no exceptions**].

ASPP14:FPT_AEX_EXT.1.2
The application shall [***not allocate any memory region with both write and execute permissions***].

ASPP14:FPT_AEX_EXT.1.3
The application shall be compatible with security features provided by the platform vendor.

ASPP14: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.

ASPP14:FPT_AEX_EXT.1.5
The application shall be built with stack-based buffer overflow protection enabled.

5.1.6.2 Use of Supported Services and APIs (ASPP14:FPT_API_EXT.1)

ASPP14:FPT_API_EXT.1.1
The application shall use only documented platform APIs.

5.1.6.3 Software Identification and Versions (ASPP14:FPT_IDV_EXT.1)

ASPP14:FPT_IDV_EXT.1.1
The application shall be versioned with [***[proprietary versioning]***]

5.1.6.4 Use of Third Party Libraries (ASPP14:FPT_LIB_EXT.1)

ASPP14:FPT_LIB_EXT.1.1
The application shall be packaged with only [**curl, openldap, apache, zlib, xerces, sqllite 3, openssl**].

5.1.6.5 Integrity for Installation and Update (ASPP14:FPT_TUD_EXT.1)

ASPP14:FPT_TUD_EXT.1.1
The application shall [***provide the ability***] to check for updates and patches to the application software.

ASPP14:FPT_TUD_EXT.1.2
The application shall [***provide the ability***] to query the current version of the application software.

ASPP14:FPT_TUD_EXT.1.3
The application shall not download, modify, replace or update its own binary code.

ASPP14:FPT_TUD_EXT.1.4
Application updates shall be digitally signed such that the application platform can cryptographically verify them prior to installation.

ASPP14:FPT_TUD_EXT.1.5
The application is distributed [***as an additional software package to the platform OS***] .

5.1.6.6 Integrity for Installation and Update - per TD0664 (ASPP14:FPT_TUD_EXT.2)

ASPP14:FPT_TUD_EXT.2.1
The application shall be distributed using the [***format of the platform-supported package manager***].

ASPP14: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

ASPP14:FPT_TUD_EXT.2.3

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

5.1.7 Trusted path/channels (FTP)

5.1.7.1 Protection of Data in Transit - per TD0743 (ASPP14:FTP_DIT_EXT.1)

ASPP14:FTP_DIT_EXT.1.1

The application shall [***encrypt all transmitted [4] with [HTTPS as a server in accordance with FCS_HTTPS_EXT.1/Server for [web server], HTTPS as a server using mutual authentication in accordance with FCS_HTTPS_EXT.2 for [web server], TLS as a server as defined in the Functional Package for TLS and also supports functionality for [mutual authentication] for [web server]]***]
between itself and another trusted IT product.

5.2 TOE Security Assurance Requirements

The SARs for the TOE are the components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

Requirement Class	Requirement Component
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_TSU_EXT.1: Timely Security Updates
ATE: Tests	ATE_IND.1: Independent Testing - Conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability Survey

Table 3 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 Basic Functional Specification (ADV_FSP.1)

ADV_FSP.1.1d

The developer shall provide a functional specification.

ADV_FSP.1.2d

The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.1.1c

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2c

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c

The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4c

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2e

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational User Guidance (AGD_OPE.1)

AGD_OPE.1.1d

The developer shall provide operational user guidance.

AGD_OPE.1.1c

The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c

The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences, and implications for maintaining secure operation.

AGD_OPE.1.6c

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1d

The developer shall provide the TOE, including its preparative procedures.

AGD_PRE.1.1c

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2e
The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d
The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1c
The application shall be labelled with a unique reference.

ALC_CMC.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.2 TOE CM Coverage (ALC_CMS.1)

ALC_CMS.1.1d
The developer shall provide a description in the TSS of how timely security updates are made to the TOE. Application developers must support updates to their products for purposes of fixing security vulnerabilities.

ALC_CMS.1.1c
The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2c
The configuration list shall uniquely identify the configuration items.

ALC_CMS.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.3 Timely Security Updates (ALC_TSU_EXT.1)

ALC_TSU_EXT.1.1d
The developer shall provide a description in the TSS of how timely security updates are made to the TOE. Note: Application developers must support updates to their products for purposes of fixing security vulnerabilities.

ALC_TSU_EXT.1.2d
The developer shall provide a description in the TSS of how users are notified when updates change security properties or the configuration of the product.

ALC_TSU_EXT.1.1c
The description shall include the process for creating and deploying security updates for the TOE software.

ALC_TSU_EXT.1.2c
The description shall express the time window as the length of time, in days, between public disclosure of a vulnerability and the public availability of security updates to the TOE.

ALC_TSU_EXT.1.3c
The description shall include the mechanisms publicly available for reporting security issues pertaining to the TOE.

Note: The reporting mechanism could include web sites, email addresses, as well as a means to protect the sensitive nature of the report (e.g., public keys that could be used to encrypt the details of a proof-of-concept exploit).

ALC_TSU_EXT.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4 Tests (ATE)

5.2.4.1 Independent Testing - Conformance (ATE_IND.1)

ATE_IND.1.1d
The developer shall provide the TOE for testing.

ATE_IND.1.1c
The TOE shall be suitable for testing.

ATE_IND.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.1.2e
The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability Survey (AVA_VAN.1)

AVA_VAN.1.1d
The developer shall provide the TOE for testing.

AVA_VAN.1.1c
The TOE shall be suitable for testing.

AVA_VAN.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence..

AVA_VAN.1.2e
The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3e
The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

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/channels

6.1 Cryptographic support

The TOE includes the Axway OpenSSL version 3.0.13 (with 3.0.8 FIPS), a cryptomodule providing the cryptographic algorithms:

Requirements	Functions	Standards	Cert
	Cryptographic key generation		
FCS_CKM_EXT.1 FCS_CKM.1/AK	RSA schemes using cryptographic key sizes of 2048, 3072, or 4096	FIPS Pub 186-4, RSA	A5302
FCS_CKM_EXT.1 FCS_CKM.1/AK	ECC schemes using 'NIST curves' P-256, P-384, P-521	FIPS Pub 186-4, ECDSA	A5302
	Password Conditioning		
FCS_CKM_EXT.1/PBKDF	Password-based key derivation with 2048 iterations to generate 256-bit keys	NIST SP 800-132	A5302
	Cryptographic key establishment/distribution		
FCS_CKM.2	RSA-based key establishment schemes	RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017	Tested with known good implementation
FCS_CKM.2	Elliptic curve-based key establishment schemes	NIST SP 800-56Ar3	A5302
	Encryption/Decryption		
FCS_COP.1/SKC	AES-CBC (128 and 256 bits) AES-GCM (128 and 256 bits)	NIST SP 800-38A NIST SP 800-38D	A5302
	Cryptographic Hashing		
FCS_COP.1/Hash	SHA-1/256/384/512 (digest sizes 160, 256, 384 and 512 bits)	FIPS Pub 180-4	A5302
	Cryptographic Signature		
FCS_COP.1/Sig	RSA schemes using cryptographic key sizes of 2048, 3072, or 4096	FIPS Pub 186-4, RSA	A5302
FCS_COP.1/Sig	ECDSA schemes using 'NIST curves' P-256, P-384, P-521	FIPS PUB 186-4, ECDSA	A5302
	Keyed-hash message authentication		
FCS_COP.1/KeyedHash	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384(key and output MAC sizes 160, 256, and 384, respectively)	FIPS Pub 198-1 FIPS Pub 180-4	A5302
	Random bit generation		
FCS_RBG_EXT.2	DRBG AES-256 CTR	NIST SP 800-90A	A5302

Table 4 OpenSSL Cryptographic Algorithms

The Axway Security Kernel supports the following CSPs:

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
AES key	Symmetric key	1. Generated using a NIST [SP 800-90A] DRBG. 2. Generated using Diffie-Hellman key agreement. 3. Derived from TLS master secret.	N/A	Plaintext in volatile memory only	Zeroized after use	Encrypt plaintext/ Decrypt ciphertext
RSA private key	Private key	Generated internally using a NIST [SP 800-90A] DRBG.	N/A	Plaintext in volatile memory only	Zeroized upon use	Decrypt ciphertext/ Sign messages (usually hash values)
RSA public key	Public key	1. Generated internally using a NIST [SP 800-90A] DRBG. Imported in plaintext form.	N/A	Plaintext in volatile memory only	Zeroized upon use	Encrypt plaintext/ Verify signatures
ECDSA private key	Private key	Generated internally	N/A	Plaintext in volatile memory only	Zeroized upon use	Sign messages (usually hash values)
ECDSA public key	Public key	1. Generated internally using a NIST [SP 800-90A] DRBG. 2. Imported in plaintext form.	N/A	Plaintext in volatile memory only	Zeroized upon use	Verify signatures
ECDH public keys *p*,*g*	Public keys	1. Generated internally using a NIST [SP 800-90A] DRBG. 2. Input in plaintext form.	N/A	Plaintext in volatile memory	Zeroized upon use	Establish symmetric keys
ECDH private keys *a*,*b*	Private key	Generated internally using a NIST [SP 800-90A] DRBG.	N/A	Plaintext in volatile memory	Zeroized upon use	Establish symmetric keys
NIST SP800-90A DRBG seed	DRBG Seed	Generated using either BcryptGenRandom on Windows, RDSEED on Linux, or /dev/urandom on Linux or Solaris.	N/A	Plaintext in volatile memory only	Zeroized when new seed is entered	Generate random numbers

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
TLS master secret	TLS master secret	1. Generated internally using a NIST [SP 800-90A] DRBG. 2. Input via TLS sessions in encrypted form	N/A	Plaintext in volatile memory only	Zeroized when TLS session is over	Derive keys in TLS sessions
HMAC Software integrity test key	Software integrity test key	Hard coded	N/A	Plaintext in hard disk and in volatile memory	Zeroized when the library integrity test is completed	Software integrity test
HMAC Key	Log Signing key	Generated internally using a NIST [SP 800-90A] DRBG.	N/A	Plaintext in volatile memory only	Zeroized when the server has signed the last log file before shutdown	Log file signing

Table 5 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs

ASPP14:FCS_CKM_EXT.1 and ASPP14:FCS_CKM.1/AK:

The TOE (VA Server) supports asymmetric key generation as described below:

- The TOE generates a TLS key that it uses to protect HTTPS communications with the Admin UI. The TOE uses RSA Keys 2048 bit or greater and ECDSA keys using NIST curves, P-256, P-384 and P-521.
- The TOE supports asymmetric key generation when the TOE generates ephemeral ECDH keys for TLS key exchange which meets the ECC scheme. The TOE uses keys with NIST curves, P-256, P-384, P-521.
- The TOE supports asymmetric key generation to create certificates that are used to sign OCSP responses. The TOE can generate ECDSA key-pairs using P-256, P-384 or P-521, or RSA with 2048, 3072 or 4096 bit key sizes.

ASPP14:FCS_CKM_EXT.1/PBKDF:

The TOE accepts a password that it uses to decrypt all private keys stored in its keystore. The TOE applies a PRF, HMAC-SHA1, along with a salt value to the input password to produce a derived key (256 bits) which can then be used as a cryptographic key in subsequent operations. The salt is generated using OpenSSL's 'rand_bytes" call, which calls the RBG.

The key is derived from the password using the OpenSSL PEM_write_bio_PrivateKey function, and the HMAC-SHA1 hash is iterated 2048 times. The password derivation algorithm used by PBKDF takes approximately 10 microseconds per iteration on a 2.33 GHZ CPU. One thousand iterations will take 10 milliseconds. 2048 iterations will take 20 milliseconds.

The TOE does not store or encode the password, but rather feeds it directly to the PBKDFv2 and uses that output directly as the encryption key.

ASPP14:FCS_CKM.2:

The TOE supports RSA key establishment schemes conforming to RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017 and Elliptic curve-based key establishment schemes conforming to NIST SP 800-56Ar3 for establishment of administrative TLS/HTTPS sessions.

ASPP14:FCS_COP.1/Hash:

The TOE uses TLS ciphersuites employing the HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-384 algorithms (which in turn utilize SHA-1, SHA-256, and SHA-384) as part of HMAC integrity protecting communications between a remote administrator and the TOE. The TOE also uses the SHA algorithms (SHA-1, SHA-256, SHA-384 or SHA-512) as part of asymmetric signature operations (when verifying CRLs and OCSP responses and generating OCSP responses).

ASPP14:FCS_COP.1/KeyedHash:

As described above, the TOE uses TLS ciphersuites employing the HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-384 algorithms as part of HMAC integrity protecting communications between a remote administrator and the TOE.

ASPP14:FCS_COP.1/Sig:

The TOE generates and verifies digital signatures in accordance with the RSA scheme using key sizes of 2048 bits and higher and the ECDSA scheme using NIST curves P-256, P-384 and P-521. Signing is done during HTTPS/TLS handshake (as the TOE acts as a server and authenticates using certificates) and when generating OCSP responses. The TOE verifies signatures during TLS mutual authentication (verifying the client's certificate message signature) and when verifying CRLs and OCSP responses.

ASPP14:FCS_COP.1/SKC:

The TOE uses AES CBC and GCM algorithms with key sizes 128 bits and 256 bits as part of HTTPS/TLS for remote administration with the Admin UI. The TLS protocol encrypts and decrypts all configuration requests that the administrator makes through the web based Admin UI and all the responses the administrator receives from the VA Server. Additionally, when configured as a responder, the VA server uses a PBKDFv2 dervied AES-CBC-256 key to encrypt its private keys.

ASPP14:FCS_HTTPS_EXT.1/Server and ASPP14:FCS_HTTPS_EXT.2:

The TOE (VA Server) implements the HTTPS protocol to provide protected communication for remote administration (the Admin WebUI). The TOE implements HTTPS according to RFC 2818 by using a TLSv1.2 session to secure the HTTP connection. When an administrator connects to the Admin WebUI using certificate based authentication, if the Admin UI cannot determine the validity and revocation status of the certificate, the TOE will not allow the administrator to connect.

ASPP14:FCS_RBG_EXT.1 and ASPP14:FCS_RBG_EXT.2
The Axway Security Kernel uses a NIST [SP 800-90A] DRBG in CTR_DRBG (AES) mode to generate cryptographic keys. The DRBG is seeded differently depending on the hardware/software platform that it is running on. On Windows, it is seeded using the windows BCryptGenRandom function. On Linux, running on processors with the RDSEED instruction, it is seeded using four successive invocations of the RDSEED instruction.

ASPP14:FCS_STO_EXT.1:

The TOE implements functionality and also invokes the functionality provided by the platform to securely store its persistent credentials including PKI private keys and passwords as follows:

- As both a Responder and Repeater, the VA server has an HTTPS/TLS server private key (adminserver.key) for secure remote administration using the Admin UI. The VA Server securely stores this private key in the entserv directory (where apache expects it) in encrypted format using AES-256.
- As a Responder, the VA Server has private keys used to sign OCSP responses which it securely stores in the vacs_db (an encrypted database file) in PKCS8/PKCS12 encrypted format using AES-256.

- On both Windows and Linux, the VA Server password, used to access the keystore (vacs.db) and to encrypt the adminserver.key, is conditioned according to ASPP14:FCS_CKM_EXT.1/PBKDF as described above.

- On Windows, the VA Server password is also encrypted by Microsoft's DPAPI and stored in the Windows registry.

PKGTLS11:FCS_TLSS_EXT.1:

The TOE (VA Server) implements TLSv1.2 in support of HTTPS communications for remote administration using the Admin WebUI. The TLS version and supporting ciphersuites are configured by the administrator such that the VA Server will deny any connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and TLS 1.1. When establishing an SSL session, the client and server use their respective cipher suite information to determine the strongest cipher suite that they have in common. This strongest cipher suite is used to exchange information.

The VA Server implementation of the TLS Server Protocol supports the following TLS ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246
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_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
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_128_CBC_SHA256 as defined in RFC 5289
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

The TOE uses either sends it RSA public key or an explicit curve (one of P-256, P-384, or P-521) as the key agreement parameters of the server's Key Exchange message if negotiating a TLS_RSA_* or TLS_ECDHE_* ciphersuite, respectively.

PKGTLS11:FCS_TLSS_EXT.2:

The VA Server supports mutual authentication of TLS clients using X.509v3 certificates. Mutual authentication of TLS clients in this case means when a remote administrator is connecting to the Admin WebUI and using a certificate for authentication. The VA Server Admin WebUI will compare the distinguished name of the certificate presented by a remote administrator against an internal list of distinguished names to ensure that the certificate is recognized.

6.2 User data protection

ASPP14:FDP_DAR_EXT.1:
The TOE does not contain any other sensitive data on the system that is not protected by FCS_STO_EXT.1. The sensitive data is protected in accordance with FCS_STO_EXT.1 as described above in Section 6.1 and consists of the VA Server password, TLS private keys and private keys used to sign OCSP responses.

ASPP14:FDP_DEC_EXT.1
The TOE does not access any hardware resources other than network connectivity and does not access any sensitive information repositories.

ASPP14:FDP_NET_EXT.1:
Network communications on the TOE are restricted to user-initiated and remote-initiated communication. The TOE requires network access for importing CRLs and checking for software updates as well as accepting incoming requests to the Admin WebUI, incoming revocation queries and outgoing revocation queries.

6.3 Identification and authentication

ASPP14:FIA_X509_EXT.1 and ASPP14:FIA_X509_EXT.2:

The TOE uses X.509 certificates for TLS/HTTPS authentication and supports CRL revocation checking. The TOE, by default, makes the Admin WebUI only available using TLS (HTTPS) and provides support for TLS mutually authenticated sessions when certificate based login is configured.

The VA server installation process automatically creates an Admin WebUI TLS certificate used within secure admin TLS connections. The TLS session encrypts and decrypts all administrative configuration requests and the corresponding responses from the TOE. Users must use their user ID and password to log in to the Admin WebUI the first time. The system presents the user with a list of certificates and requests the user to select the certificate to use for subsequent logins. If no certificates are found, the user is requested to browse to the certificate that will be used for login. The server then validates the certificate. When the user logs in again, the user will only have to enter his user ID, as long as his login certificate remains valid on the server.

For TLS mutually authenticated sessions, the TOE validates the client certificate during connection establishment. First, the TOE validates that it can construct a certificate path from the certificate through any intermediary CAs to a configured trusted root CA. The VA Server discovers all the digital certificates in the certification path, using the issuer name hash, the AIA information, and the subject name, and obtains associated CRL data in real time, as needed, to perform the validation. If the path can be constructed, the validity date and CA flag is checked in each CA certificate. If all of those checks succeed, the TOE finally checks the revocation status using a configured CRL of all certificates in the path. The TOE operates in an unattended mode and will reject any certificate for which it cannot determine validity and will reject the connection attempt.

The TOE will check for the correct CA flag in the basicConstraints extension when a CA is uploaded to the truststore. Any CA certificate that does not have the CA flag set to true in the basicConstraints extension will be rejected.

6.4 Security management

ASPP14:FMT_CFG_EXT.1:

The TOE provides the ability to set new credentials during the installation process. The TOE provides a web-based administration server that provides centralized management of its validation processing components, including the TOE security functions, through an Admin UI. The Admin UI can be accessed remotely from a browser over a secure HTTPS connection. Only authorized administrators can access the Admin UI, and must do so by entering either a password or by having a valid certificate authorized for VA administration server access. The VA server installs with a default user name ‘admin’ but no default password. When installing the VA Server for the first time, a password must be specified, which, after the installation completes, is used to access the Admin UI **SETUP** menu for initial configuration. The installation also automatically creates a VA administration server private key (adminserver.key) and SSL certificate (adminserver.crt) in the<*VAInstallDir*>\entserv directory. Additional administrator accounts can be created with either a user ID and password or certificate.

ASPP14:FMT_MEC_EXT.1:

The TOE invokes mechanisms on its platform for storing and setting configuration options. The configuration data for the TOE on Windows is stored in the Windows Registry as well as the ProgramData directory. For the TOE on Linux, the configuration data is stored in the /etc directory (for system specific configuration) or in the user’s home directory (for user specific configuration).

ASPP14:FMT_SMF.1:

The TOE performs the following security management functions.

Configure Users & Password Policy

- User accounts can be created via the following setting in the Admin UI: *CONFIGURATION menu, click User Settings > User Accounts*

- General user account settings (e.g. password policy) can be configured via the following setting in the Admin UI: *CONFIGURATION menu, click User Settings > General Settings*

Configure Certificates

The VA Server has two basic types of certificate stores: a CA Certificate store for certificates issued by a Certificate Authority and a VA Server Certificate Store for certificates issued by the VA Server.

- Certificates can be viewed, added or modified via the following setting in the Admin UI: *CONFIGURATION menu, click Keys and Certificates > Certificates*

- Certificate requests can be created via the following setting in the Admin UI: *CONFIGURATION menu, click Keys and Certificates > click Create/Import Private Key*

- The VA server installation process automatically creates an Admin UI TLS certificate to provide a secure connection between the browser and remote administrators using Transport Layer Security (TLS). The VA administration server TLS private key (`adminserver.key`) is placed in the `<VAInstallDir>/entserv` directory.

Configure Revocation Sources

- Selection and configuration of sources from which the VA Server obtains certificate revocation lists can be configured via the following setting in the Admin UI: *CONFIGURATION > CRLs > CRL Import*.

- As a Responder, Certificate Authority (CA) options can be configured via the following setting in the Admin UI: *CONFIGURATION menu, click Server Settings > CA options*

- Server URLs can be configured via the following setting in the Admin UI: *CONFIGURATION menu, click Server Settings > Server URLs*

Create/Import Key Pair

- Key pairs can be generated or imported via the following setting on the Admin UI: *CONFIGURATION menu, click Keys and Certificates > Create/Import Private Key*

Check for Software Updates

- From the Help menu, click *Check for Updates* to display the latest available version / build of Validation Authority Server

6.5 Privacy

ASPP14:FPR_ANO_EXT.1:

The TOE does not collect personally identifiable information (PII) for administrators or users and, therefore, does not transmit any PII over a network.

6.6 Protection of the TSF

ASPP14:FPT_AEX_EXT.1:

The TOE protects itself against exploitation by implementing address space layout randomization (ASLR) and by not allocating any memory region for both write and execute permission. The TOE is compiled for both Windows and Linux with stack-based buffer overflow protection as follows:

- Windows: The /GS flag was used during compilation to enable ASLR and stack-based buffer overflow protection.
- Linux: Since the TOE software is compiled using GCC, the fstackprotector-all flag was used during compilation to enable ASLR and stack-based buffer overflow protection.

Additionally, by default, the TOE does not allow user-modifiable files to be written to directories that contain executable files.

ASPP14:FPT_API_EXT.1:

The TOE only uses documented platform APIs from Microsoft Windows C/C++ SDK and APIs from Linux GNU C Library (glibc). The TOE utilizes the following Windows APIs:

- WinSock
- PSAPI
- BCrypt
- WNetAPI
- C Library API
- WinHTTP
- CryptoAPI
- WinINet
- Native Windows API
- COM
- IPHelper
- Active Directory Services
- ODBC API
- RPC
- Ntsecapi
- Windows API which consists of:
  - Kernel32.dll – for basic services
  - advapi32.dll – for advanced services
  - gdi32.dll – for Graphiscs Device Interface
  - user32.dll - for User Interface
  - comdlg32.dll – For Common Dialog Box
  - shell32.dll & shlwapi.dll – For Windows Shell
  - ole32.dll & oleaut32.dll – for Object Linking and Embedding

The TOE supports the following Linux APIs:

- POSIX thread API
- Libc
- librt for real time extensions
- libresolv.

ASPP14:FPT_IDV_EXT.1:

The TOE utilizes a software version with a major version with a minor update and build number.

ASPP14:FPT_LIB_EXT.1:

The TOE includes a number of third party libraries used to perform its functions as identified in the table below:

3rd Party Library	Version	Function
Curl	8.7.1	Retrieving CRLs
Openldap	2.6.4	Retrieving Certificates, CRLs via LDAP
Apache httpd	2.4.59	Web UI configuration interface hosting
Zlib	1.2.13	Data compression/decompression
Xerces	3.2.5	XML parsing

SQLite3	3.44.2	CRL database, VA Certificate Store database and VA Configuration Database
Openssl	3.0.13	OCSP / SCVP requests, responses, encryption, decryption, signing, verification.

ASPP14:FPT_TUD_EXT.1/ASPP14:FPT_TUD_EXT.2:

The TOE includes mechanisms to check for updates and to query the current version of the application software. The VA Server displays the current version of the TOE via the Admin UI, and the VA Server also provides a 'check for updates' button in the Admin WebUI for the administrator to check for updates.

TOE software is digitally signed and distributed using the platform-supported package manager (Windows or Linux). A signing certificate issued by Digicert is used to sign the installation package for Windows platforms. For Linux platforms, the RPM package is signed with the Axway GPG key which the administrator has to add into the list of trusted RPM signing keys. The TOE does not update its own binary code in any way and when removed, all traces of the TOE application software are deleted.

Axway addresses all vulnerabilities found in the product within 30 days from public disclosure. Users can report any security issues pertaining to the TOE by contacting Axway's technical support via phone or email. The phone number and email address are published on the Axway support website and provided directly to clients. TOE updates are not posted publicly and are only provided to customers who have contracts with Axway. When any changes are made to the product, whether security related or not, users will receive a message from Axway informing them that there is an update available. The updates are deployed to Axway's software repository from which users can download the updates.

6.7 Trusted path/channels

ASPP14:FTP_DIT_EXT.1:

The TOE (VA Server) provides a trusted path for its remote administrative users accessing the Admin UI using TLS/HTTPS.