Axway Validation Authority Suite (ASPP12) Security Target

Version 1.0
08/01/2019

Prepared for:

Axway, Inc.

6811 E Mayo Blvd, Ste 400,
Phoenix, AZ 85054

Prepared By:

Gossamer Laboratories

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 the Axway Validation Authority Suite provided by Axway, Inc. The TOE is being evaluated as an Application Software against the Protection Profile for Application Software, Version 1.2, 22 April 2016, hereafter referred to as ASPP12 in this ST.

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 Suite (ASPP12) Security Target

ST Version – Version 1.0

ST Date – 08/01/2019

1.2 TOE Reference

TOE Identification – Axway Inc., Axway Validation Authority Suite, Version 5.0 comprises the following software components:

- Validation Authority Server

- Desktop Validator (Standard & Enterprise Editions)
- Server Validator

TOE Developer – Axway, Inc.

Evaluation Sponsor – Axway, Inc.

1.3 TOE Overview

The Target of Evaluation (TOE) is the Axway Validation Authority Suite, Version 5.0.

1.4 TOE Description

The Axway Validation Authority (VA) Suite 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 Axway VA Suite 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.[1] 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.[2]

For purposes of this evaluation, the Axway VA Suite is a software application that offers CAVP certified 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 Suite consists of three software components, the Validation Authority Server, the Desktop Validator (Standard and Enterprise Editions) and the Server Validator, which run on one of the following platforms:

- Microsoft Windows 2012 (64 bit) on a 64 bit Intel Xeon processor
- RHEL 7 (64 bit) on a 64 bit Intel Xeon processor

The Windows and RHEL platforms are part of the operating environment of the TOE. The Axway VA Suite applications and the particular platforms they each run on are described below:

[1] The generation and signing of OCSP/SCVP has not been tested in the evaluated configuration.
[2] The use of a Hardware Security Module (HSM) is not included in the evaluated configuration.

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 realtime 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.
3. Server Validator (SV) – the Server Validator provides revocation status checking of client certificates for web servers. SV serves as an interface between a secure web server and a VA Server. The SV runs on Windows and Linux. Web servers requiring SSL and client authentication can use the Server Validator to add revocation checking for certificates to authenticate to web servers or web-enabled applications.

The cryptographic capabilities of Axway VA Suite are provided by the Axway Security Kernel and are maintained across all three TOE software applications. The Axway Security Kernel (version 3.0.2) 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 upon a custom version of OpenSSL 1.0.2k. The implementation of TLS/HTTPS to secure communication channels is supported using Apache.

The TOE components of the Axway VA Suite all share the same underlying cryptographic library and can be deployed together to provide a robust and flexible certificate validation solution that covers both desktop and web enabled applications.

1.4.1.1 Physical Boundaries

The TOE is composed of three software-only applications which execute 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 ASPP12.

Specifically, the evaluated configuration includes the following:

- Axway Validation Authority Server v5.0 - a software server application running on the following two platforms:
  - Microsoft Windows 2012 (64 bit) on a 64 bit Intel Xeon processor
  - RHEL 7 (64 bit) on a 64 bit Intel Xeon processor
- Axway Desktop Validator (Standard & Enterprise Editions)[3] v5.0 – a software client application running on the following platform:
  - Microsoft Windows 2012 (64 bit) on a 64 bit Intel Xeon processor
- Axway Server Validator v5.0 – a software client application running on the following two platforms:
  - Microsoft Windows 2012 (64 bit) on a 64 bit Intel Xeon processor
  - RHEL 7 (64 bit) on a 64 bit Intel Xeon processor

Server Validator provides revocation checking for the following web servers in the operational environment: Apache 2.4 and Oracle HTTP Server (OHS) 12c.

[3] The Enterprise Edition of the Desktop Validator was tested in the evaluated configuration.

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

1.4.1.2 Logical Boundaries

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

- 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 by the Axway Security Kernel, a cryptographic module built upon a custom version of OpenSSL 1.0.2k, 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 or sensitive information repositories and no sensitive data is stored in non-volatile memory. Inbound and outbound network communications are restricted to those that are user-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.

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:

- Axway Validation Authority Version 5.0 Common Criteria Guide, 31 July 2019

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 1.2, 22 April 2016 (**ASPP12**) with the following NIAP Technical Decisions applied: TD0119, TD0131, TD0163, TD0178, TD0215, TD0217, TD0238, TD0241, TD0267, TD0268, TD0295, TD0296, TD0300, TD0326, TD0327, TD0358, TD0359, TD0380, TD0382, TD0389, TD0390 and TD0427.

2.1 Conformance Rationale

The ST conforms to the ASPP12. 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 ASPP12 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The ASPP12 offers additional information about the identified security objectives, but that has not been reproduced here and the ASPP12 should be consulted if there is interest in that material.

In general, the ASPP12 has defined Security Objectives appropriate for application software and as such are applicable to the Axway Validation Authority Suite 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 systemand 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 ASPP12. The ASPP12 defines the following extended requirements and since they are not redefined in this ST the ASPP12 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

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

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 ASPP12. The refinements and operations already performed in the ASPP12 are not identified (e.g., highlighted) here, rather the requirements have been copied from the ASPP12 and any residual operations have been completed herein. Of particular note, the ASPP12 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 ASPP12 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the ASPP12 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The ASPP12 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

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

Requirement Class	Requirement Component
FCS: Cryptographic support	FCS_CKM.1(1): Cryptographic Asymmetric Key Generation
	FCS_CKM.1(A): Cryptographic Key Derivation
	FCS_CKM.2: Cryptographic Key Establishment
	FCS_CKM_EXT.1: Cryptographic Key Generation Services
	FCS_COP.1(1): Cryptographic Operation - Encryption/Decryption
	FCS_COP.1(2): Cryptographic Operation - Hashing
	FCS_COP.1(3): Cryptographic Operation - Signing
	FCS_COP.1(4): Cryptographic Operation - Keyed-Hash Message Authentication
	FCS_HTTPS_EXT.1: HTTPS Protocol
	FCS_RBG_EXT.1: Random Bit Generation Services
	FCS_RBG_EXT.2: Random Bit Generation from Application
	FCS_STO_EXT.1: Storage of Credentials
	FCS_TLSS_EXT.1: TLS Server Protocol
FDP: User data protection	FDP_DAR_EXT.1: Encryption Of Sensitive Application Data
	FDP_DEC_EXT.1: Access to Platform Resources
	FDP_NET_EXT.1: Network Communications
FIA: Identification and authentication	FIA_X509_EXT.1: X.509 Certificate Validation
	FIA_X509_EXT.2: X.509 Certificate Authentication
FMT: Security management	FMT_CFG_EXT.1: Secure by Default Configuration
	FMT_MEC_EXT.1: Supported Configuration Mechanism
	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_AEX_EXT.1: Anti-Exploitation Capabilities
	FPT_API_EXT.1: Use of Supported Services and APIs
	FPT_LIB_EXT.1: Use of Third Party Libraries
	FPT_TUD_EXT.1: Integrity for Installation and Update
FTP: Trusted path/channels	FTP_DIT_EXT.1: Protection of Data in Transit

Table 1 TOE Security Functional Components

5.1.1 Cryptographic support (FCS)

5.1.1.1 Cryptographic Asymmetric Key Generation (FCS_CKM.1(1))

FCS_CKM.1.1(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-256, P-384 and [P-521]] that meet the following: [FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4],***

(TD0326 applied)

5.1.1.2 Cryptographic Key Derivation (FCS_CKM.1(A))

FCS_CKM.1.1(A)

Refinement: A password/passphrase shall perform [Password-based Key Derivation Functions] in accordance with a specified cryptographic algorithm as specified in FCS_COP.1(4), with [**2048**] iterations, and output cryptographic key sizes [***256***] that meet the following: [NIST SP 800-132]. (applied per TD0119)

FCS_CKM.1.2(A)

The TSF shall generate all salts using a RBG that meets FCS_RBG_EXT.1 (from the AS PP) and with entropy corresponding to the security strength selected for PBKDF in FCS_CKM.1.1(A). (applied per TD0119)

5.1.1.3 Cryptographic Key Establishment (FCS_CKM.2)

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 meets the following: [NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography"],***
- ***[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"],***

(TD0326 applied)

5.1.1.4 Cryptographic Key Generation Services (FCS_CKM_EXT.1)

FCS_CKM_EXT.1.1

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

5.1.1.5 Cryptographic Operation - Encryption/Decryption (FCS_COP.1(1))

FCS_COP.1.1(1)

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

5.1.1.6 Cryptographic Operation - Hashing (FCS_COP.1(2))

FCS_COP.1.1(2)

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.

5.1.1.7 Cryptographic Operation - Signing (FCS_COP.1(3))

FCS_COP.1.1(3)

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 4,***
- ***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 5***].

5.1.1.8 Cryptographic Operation - Keyed-Hash Message Authentication (FCS_COP.1(4))

FCS_COP.1.1(4)

The application shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-SHA-256 and [***SHA-1, SHA-384, SHA-512***] with key sizes [***160, 256, 384, 512***] and message digest sizes 256 and [***160, 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.

5.1.1.9 HTTPS Protocol (FCS_HTTPS_EXT.1)

FCS_HTTPS_EXT.1.1

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

FCS_HTTPS_EXT.1.2

The application shall implement HTTPS using TLS in accordance with ***[FCS_TLSS_EXT.1]***. (TD0215 applied)

FCS_HTTPS_EXT.1.3

The application shall [***notify the user and not establish the connection***] if the peer certificate is deemed invalid. (TD0296 applied)

5.1.1.10 Random Bit Generation Services (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1

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

5.1.1.11 Random Bit Generation from Application (FCS_RBG_EXT.2)

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)***].

FCS_RBG_EXT.2.2

The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [***no other 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.

5.1.1.12 Storage of Credentials (FCS_STO_EXT.1)

FCS_STO_EXT.1.1

The application shall [***invoke the functionality provided by the platform to securely store*** [**passwords**], ***implement functionality to securely store*** [**private keys, passwords**]] to non-volatile memory.

5.1.1.13 TLS Server Protocol (FCS_TLSS_EXT.1)

FCS_TLSS_EXT.1.1

The application shall [***implement TLS 1.2 (RFC 5246)***] supporting the following 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 no other ciphersuite.
(TD0358 applied)

FCS_TLSS_EXT.1.2

The application shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1, and [***none***].

FCS_TLSS_EXT.1.3

The application shall generate key establishment parameters using [***RSA with key size*** [***2048 bits***, ***3072 bits, 4096 bits***], ***ECDHE over NIST curves*** [***secp256r, secp384r***] ***and no other curves***]. (TD0326 applied)

FCS_TLSS_EXT.1.4

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

FCS_TLSS_EXT.1.5

The application shall not establish a trusted channel if the peer certificate is invalid.

FCS_TLSS_EXT.1.6

The application shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

5.1.2 User data protection (FDP)

5.1.2.1 Encryption Of Sensitive Application Data (FDP_DAR_EXT.1)

FDP_DAR_EXT.1.1

The application shall [***protect sensitive data in accordance with FCS_STO_EXT.1***] in non-volatile memory. (TD0300 applied)

5.1.2.2 Access to Platform Resources (FDP_DEC_EXT.1)

FDP_DEC_EXT.1.1

The application shall restrict its access to [***no hardware resources***].

FDP_DEC_EXT.1.2

The application shall restrict its access to [***no sensitive information repositories***].

5.1.2.3 Network Communications (FDP_NET_EXT.1)

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 initiated communication for incoming requests to the Admin UI and incoming revocation queries], [application-initiated communication for outgoing revocation queries]***]

5.1.3 Identification and authentication (FIA)

5.1.3.1 X.509 Certificate Validation (FIA_X509_EXT.1)

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 and that the CA flag is set to TRUE for all CA certificates.

The application shall validate the revocation status of the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5759***]. (TD0217 applied)

The application shall validate the extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage 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 extendedKeyUsage field.

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 (FIA_X509_EXT.2)

FIA_X509_EXT.2.1

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

FIA_X509_EXT.2.2

When the application cannot establish a connection to determine the validity of a certificate, the application shall [***not accept the certificate***].

5.1.4 Security management (FMT)

5.1.4.1 Secure by Default Configuration (FMT_CFG_EXT.1)

FMT_CFG_EXT.1.1

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

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. (TD0327 applied)

5.1.4.2 Supported Configuration Mechanism (FMT_MEC_EXT.1)

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 (FMT_SMF.1)

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions [**/**

- **Configure revocation sources**
- **Configure Certificates**
- **Create/Import Key pair**
- **Configure Users**
- **Configure Password Policy**
- **Check for Software Updates**].

5.1.5 Privacy (FPR)

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

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 (FPT_AEX_EXT.1)

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.

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

FPT_API_EXT.1.1

The application shall use only documented platform APIs.

5.1.6.3 Use of Third Party Libraries (FPT_LIB_EXT.1)

FPT_LIB_EXT.1.1

The application shall be packaged with only [***curl, openldap, apache, zlib, xerces, sqllite 3, net snmp, openssl***].

5.1.6.4 Integrity for Installation and Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

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

FPT_TUD_EXT.1.2

The application shall be distributed using the format of the platform-supported package manager.

FPT_TUD_EXT.1.3

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.1.4

The application shall not download, modify, replace or update its own binary code.

FPT_TUD_EXT.1.5

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

FPT_TUD_EXT.1.6

The application installation package and its updates 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 (FTP_DIT_EXT.1)

FTP_DIT_EXT.1.1

The application shall [***encrypt all transmitted sensitive data with [HTTPS, TLS]***] between itself and another trusted IT product. (TD0389 applied)

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 2 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 fulfil 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.1.1 d

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.1.1 d

The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1c

The TOE shall be labelled with its 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 configuration list for the TOE.

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.1.1 d

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_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. 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's Cryptographic Library, Axway Security Kernel version 3.0.2, provides the following CAVP certified algorithms:

Requirements	Functions	Standards	Cert
	Cryptographic key generation		
FCS_CKM.1(1)	RSA schemes using cryptographic key sizes of 2048-bit or greater	FIPS Pub 186-4, RSA	**RSA 2442** **C 847**
FCS_CKM.1(1)	ECC schemes using 'NIST curves' P-256, P-384, P-521	FIPS Pub 186-4, ECDSA	**ECDSA 1089** **C 847**
	Cryptographic key establishment/distribution		
FCS_CKM.2	RSA-based key establishment schemes	Vendor affirm 800-56B SHS DRBG RSA	**N/A**
FCS_CKM.2	Elliptic curve-based key establishment schemes	NIST SP 800-56A CVL KAS ECC	**Component 1177** **Component 1179** **C 847**
	Encryption/Decryption		
FCS_COP.1(1)	AES-CBC (128 and 256 bits) AES-GCM (128 and 256 bits)	NIST SP 800-38A NIST SP 800-38D	**AES 4466** **C 847**
	Cryptographic Hashing		
FCS_COP.1(2)	SHA-1/256/384/512 (digest sizes 160, 256, 384 and 512 bits)	FIPS Pub 180-4	**SHS 3678** **C 847**
	Cryptographic Signature		
FCS_COP.1(3)	RSA schemes using cryptographic key sizes of 2048-bit or greater	FIPS Pub 186-4, RSA	**RSA 2442** **C 847**
FCS_COP.1(3)	ECDSA schemes using 'NIST curves' P-256, P-384, P-521	FIPS PUB 186-4, ECDSA	**ECDSA 1089** **C 847**
	Keyed-hash message authentication		

Requirements	Functions	Standards	Cert
FCS_COP.1(4)	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 (key and output MAC sizes 160, 256, 384 and 512, respectively)	FIPS Pub 198-1 FIPS Pub 180-4	**HMAC 2964** **C 847**
	Random bit generation		
FCS_RBG_EXT.2	DRBG AES-256 CTR	NIST SP 800-90A NIST SP 800-57	**DRBG 1449** **C 847**

Table 3 OpenSSL Cryptographic Algorithms

FCS_RBG_EXT.1 and FCS_RBG_EXT.2
The Axway Security Kernel uses a NIST [SP 800-90] DRBG in CTR_DRBG(AES) mode to generate cryptographic keys. This RNG is a FIPS 140-2 approved RNG as specified in Annex C to FIPS PUB 140-2. 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. On Linux running on processors without the RDSEED instruction, it is seeded using a call to /dev/urandom.

The Axway Security Kernel supports the following CSPs:

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
TDES keys	Symmetric key	1. Generated internally using a NIST [SP 800-90] 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
AES key	Symmetric key	1. Generated using a NIST [SP 800-90] 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 800-90] 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 800-90] DRBG. Imported in plaintext form.	N/A	Plaintext in volatile memory only	Zeroized upon use	Encrypt plaintext/ Verify signatures

Key	Key Type	Generation / Input	Output	Storage	Zeroization	Use
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 800-90] 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 800-90] 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 800-90] 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
TLS master secret	TLS master secret	1. Generated internally using a NIST [SP 800-90] 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 800-90] DRBG.	N/A	Plaintext in volatile memory only	Zeroized when the server has signed the last log file before shutdown	Log file signing

Table 4 – List of Cryptographic Keys, Cryptographic Key Components, and CSPs

RSA and ECDSA public keys are output from and input into the kernel in plaintext form. Symmetric keys are input into and output from the kernel in encrypted form. Session keys are stored in volatile memory in plaintext. RSA and ECDSA key pairs are stored in hard disk in plaintext. Keys are zeroized when they are no longer used; RSA and ECDSA key pairs are zeroized when new ones are generated. The zeroization of the keys is carried out by overwriting the storage or memory with zeros. Zeroization may be manually invoked by rebooting the computer on which the kernel is running. Uninstalling the TOE software application also results in zeroization of all keys and other CSPs.

FCS _CKM.1(1) and FCS_CKM_EXT.1

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

- The VA server 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 VA Server 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 and P-521.

FCS_CKM.1(A)

The TOE (VA Server) 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 PBKDF 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 Core I7. One thousand iterations will take 10 milliseconds. 2048 iterations will take 20 milliseconds.

The password is not encoded, but rather it is fed directly to the SHA algorithm via the Openssl function 'PKCS5_PBE_keyivgen'. The output of the hash function is used directly as the encryption key.

FCS_CKM.2

The TOE (VA Server) supports RSA key establishment schemes conforming to NIST SP 800-56B and Elliptic curve-based key establishment schemes conforming to NIST SP 800-56A for establishment of TLS/HTTPS communications.

FCS_COP.1(1)

The TOE (VA Server) uses AES CBC and GCM algorithms with key sizes 128 bits and 256 bits as part of HTTPS for remote administration with the Admin UI. SSL 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 the VA server is configured as a responder, it uses AES-CBC-256 to encrypt its private keys.

FCS_COP.1(2)

The TOE (VA Server) uses TLS ciphersuites which include the SHA-128, SHA-256 and SHA-384 algorithms and are used to protect communication between a remote administrator and the TOE. The VA Server, DV and SV use the SHA algorithms (SHA-1, SHA-256, SHA-384 or SHA-512) when downloading and verifying CRLs.

FCS_COP.1(3)

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 authentication. Signature verification is performed by the VA Server during TLS mutual authentication and by the VA Server, DV and SV when verifying CRLs.

FCS_COP.1(4)

The TOE implements keyed hash message authentication in the TLS ciphersuites to protect message integrity of HTTPS/TLS communication.

FCS_HTTPS_EXT.1

The TOE (VA Server) implements the HTTPS protocol to provide protected communication for remote administration using the Admin UI. When an administrator connects to the Admin UI using certificate based

authentication, if the Admin UI cannot determine the revocation status of the certificate it will not allow the administrator to connect.

FCS_STO_EXT.1

The TOE (VA Server) 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 in encrypted format using AES-256.
- On both Windows and Linux, the VA Server password, used to access the keystore (vack.db) and to encrypt the adminserver.key, is conditioned according to FCS_CKM.1(A) as described above.
- On Windows, the VA Server password is also encrypted by Microsoft's DPAPI and stored in the Windows registry.

FCS_TLSS_EXT.1

The TOE (VA Server) implements TLSv1.2 in support of HTTPS communications for remote administration using the Admin UI. 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 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 UI and using a certificate for authentication. The VA Server Admin UI will compare 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

The TOE (VA Server, DV and SV) does not access any hardware resources or sensitive information repositories. Sensitive data consists of the VA Server password and TLS private keys. The sensitive data is protected in accordance with FCS_STO_EXT.1 as described above in Section 6.1. Network communications on the TOE are restricted to user-initiated and remote-initiated communication.

The TOE requires network access for importing CRLs, checking for software updates, incoming requests to the Admin UI, incoming revocation queries and outgoing revocation queries.

Specifically, the TOE requires network access as indicated below:

- VA Server – Import CRLs, Allow incoming requests to the Admin UI and Incoming Revocation Queries, Check for Software Updates
- DV – Outgoing Revocation Queries, Check for Software Updates
- SV- Outgoing Revocation Queries, Check for Software Updates

The User data protection function is designed to satisfy the following security functional requirements:

- 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.
- FDP_DEC_EXT.1 – the TOE does not access any hardware resources or sensitive information repositories.
- FDP_NET_EXT.1 – the TOE restricts network communications to the following: user-initiated communication for importing CRLs and checking for updates, remotely initiated communication for incoming requests to the Admin UI and incoming revocation queries, application-initiated communication for outgoing revocation queries.

6.3 Identification and authentication

The TOE (VA Server) uses X.509 certificates for TLS/HTTPS authentication and supports CRL revocation checking. The Admin UI is, by default, 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 UI TLS certificate to provide a secure connection between the browser and remote administrators using Transport Layer Security (TLS). TLS encrypts and decrypts all configuration requests that are made and all the responses from the VA server. Users must use their user ID and password to log in to the Admin UI 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 Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_X509_EXT.1: CRL is supported for X509v3 certificate validation. Certificates are validated as part of the authentication process when they are presented to the TOE and when they are loaded into the TOE.
- FIA_X509_EXT.2: The TOE supports the use of X509v3 certificates to support authentication for HTTPS/TLS. The TOE will reject any certificate for which it cannot determine validity and will reject the connection attempt.

6.4 Security management

The TOE (VA Server) 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.

The TOE (DV) requires administrative privileges on the Windows machine that it is to be installed on. During installation, the administrator enters user information and company name. The DV software is automatically installed for all users on the system. However, users without Administrator privileges will have a read-only view of configuration options, and will not be able to make any modifications to the configuration options set by the Administrator. The DV provides a configuration application which can be accessed via a desktop shortcut or by using the Start Menu. The Desktop Validator Configuration application allows the administrator to view and configure the TOE security management functions.

The TOE (SV) also requires administrative privileges on the Windows or Linux machine that it is to be installed on. Additionally, prior to installing a Server Validator (SV), an appropriate web server must be installed and configured for certificate authentication. During installation, the administrator enters user information and company name and selects the web servers where validation is to be enabled. The SV does not provide a user interface for creating or modifying its configuration. An SV instance is configured by editing the configuration file serverFilename.ini, where serverFilename is the name of the configuration file used for a particular web server. The default serverFilename.ini location is: ***serverHomeDirectory*\<*configDirectory*>*valicert**serverFilename.ini*.** Administrators can modify the configuration file using a text editor.

The configuration data for the VA and DV on Windows is stored in the Windows Registry. For the VA 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). The configuration data for the SV is stored on Windows in C:\ProgramData\Axway\SV and on Linux in the /etc/sv directory.

The TOE (VA Server, DV and SV) performs the following security management functions.

Configure Users & Password Policy

VA Server:

- 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

VA Server:

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 > Certificates*

- 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

VA Server:

- 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*

DV:

The Desktop Validator (DV) Standard edition provides certificate validation support for client applications on Microsoft Windows platforms. The Enterprise edition provides certificate validation support for both client and server applications on Microsoft Windows platforms. Desktop Validator Enterprise is required for use of server applications such as Domain Controllers, IIS, and so on.

- From the *General tab*, select the *Use Axway DV as CAPI revocation provider* option to enable digital certificate validation using Desktop Validator.
- From the *General tab*, select the *VACRL* validation protocol to configure communication with the VA Server in order to validate a certificate

SV:

- The SV is configured to use the VA Server by setting the <*EVA CERT STORE SECTION*> in the SV configuration file serverFilename.ini:

Create/Import Key Pair

VA Server:

- 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

VA Server:

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

DV:

- From the *General menu*, click *Update Check* or *Check Now* to display the latest available version / build of Desktop Validator.

SV:

- Use the *updatecheck* (updatecheck.exe in Windows) commandline utility to check for the latest available version / build of Server Validator.

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

- FMT_CFG_EXT.1: The TOE installs with a default user name ‘admin’ but no default password. The user is required to set the password when they install the TOE.

- FMT_MEC_EXT.1: The TOE invokes mechanisms on its platform for storing and setting configuration options.
- FMT_SMF.1: The TOE is capable of performing the following management functions: configure revocation sources, configure users, configure password policy, configure certificates, create/import key pair and check for software updates.

6.5 Privacy

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

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

- FPR_ANO_EXT.1: The TOE does not collect or transmit any PII over a network.

6.6 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 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.

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

The TOE supports 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.

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

Third Party Library	Version	Function	TOE Component (VA Server, SV, DV)
curl	7.64.0	Retrieving CRLs	VA Server, SV, DV
openldap	2.4.47	Retrieving Certificates, CRLs via LDAP	VA Server
apache	2.4.38	UI hosting	VA Server, DV
zlib	1.2.11	Data compression/decompression	VA Server, DV, SV
xerces	3.2.2	XML parsing	VA Server, DV, SV
Sqllite 3	3.26.0	CRL database	VA Server
net-snmp	5.8	VA status	VA Server
openssl	1.0.2k	Encryption, decryption, signing, verification.	VA Server, DV, SV

FPT_TUD_EXT.1

The TOE (VA Server, DV and SV) includes mechanisms to check for updates and to query the current version of the application software. Both the VA Server and the DV display the current version of the TOE via the Admin UI and the DV Configuration Application, respectively. The VA Server and DV also provide a 'check for update' button via their respective UIs for the administrator to check for updates. On the SV, the administrator has to call an executable (updatecheck.exe) to display the current version of the software and 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 Thawte, 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.

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

- 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 enabled.
- FPT_API_EXT.1: The TOE only uses documented platform APIs.
- FPT_LIB_EXT.1: The TOE includes a number of third party libraries used to perform its functions.
- FPT_TUD_EXT.1: The TOE provides the ability to query the current version of its application software, to check for updates to its software and to digitally sign the application installation package and its updates. The TOE is distributed using the format of the platform-supported package manager. It does not update its own binary code in any way and when removed, all traces of the TOE application software are deleted.
- ALC_TSU_EXT.1: The developer ensures that timely security updates are made to the TOE and that users are notified when security updates are available.

6.7 Trusted path/channels

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

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- FTP_DIT_EXT.1: The TOE uses HTTPS/TLS to protect communications between itself and remote administrators accessing the Admin UI from a web browser.