Venafi Trust Protection Platform Security Target

Acumen Security, LLC.

Document Version: 4.0

Table Of Contents

Revision History

Version	Date	Description
4.0	August 2020	Initial IAR release

1 Security Target Introduction

1.1 Security Target and TOE Reference

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

Category	Identifier
ST Title	Venafi Trust Protection Platform Security Target
ST Version	4.0
ST Date	August 2020
ST Author	Acumen Security, LLC.
TOE Identifier	Venafi Trust Protection Platform
TOE Software Version	20.1
TOE Developer	Venafi
Key Words	Software

Table 1 TOE/ST Identification

1.2 TOE Overview and Description

Venafi Trust Protection Platform secures and protects keys and certificates in the datacenter, on desktops, on mobile and IoT devices, and in the cloud. This protection improves security posture with increased visibility, threat intelligence, policy enforcement, and faster incident response for certificate-related outages and compromises leveraging misused keys and certificates.

The platform supports all Venafi products and provides native integration with thousands of applications and common APIs for the extensive security ecosystem. Shared and extensible services enable enterprises to gain complete visibility into their key and certificate inventory, identify certificate reputation, and establish a baseline. The entire issuance and renewal process can be automated with policy enforcement and workflows, enabling new encryption dependent applications to be scaled quickly. Trust Protection Platform keeps organizations secure, helping them comply with standards and remediate key and certificate misuse.

The description above provides a general description of the functionality provided by the Venafi Trust Protection Platform. Please see sections 1.3.2.1 through 1.3.2.6 for an identification of the evaluated functionality and section 1.3.2.7 for an identification of the functionality that is not covered by the evaluation.

1.3 TOE Architecture

1.3.1 Physical Boundaries

The TOE boundary is the application software which runs on the host platform. For this evaluation the TOE runs on Windows Server 2012 R2 configured in FIPS mode. The Universal C Runtime must be installed. In addition to this the following Microsoft Internet Information (IIS) web server roles must be installed:

- Common HTTP Features\Static Content

- Common HTTP Features\Default Document
- Health and Diagnostics\HTTP Logging
- Health and Diagnostics\Logging Tools
- Health and Diagnostics\Request Monitor
- Health and Diagnostics\Tracing
- Security\Request Filtering
- Performance\Static Content Compression

It should be noted that this operating system is outside the TOE boundary.

The following third-party libraries come bundled with the TOE and are inside the TOE boundary.

- JSON.Net
- PDFSharp
- PFDSharp Charting
- MigraDoc
- MigraDoc.Rendering
- MigraDoc.RTFRendering
- HTMLAgility Pack
- Anti-Cross Site Scripting Library
- IronPython
- jQuery
- Moment JS
- Backbone JS
- Twitter bootstrap Apache v2
- Underscore
- Boost
- Beast
- JSON11
- Base64
- Cxxopts
- Chaos.NaCl

The TOE also uses an external database to store credentials, certificates, keys and log data. Microsoft SQL Server 2012 SP2 is used in the evaluated configuration and Microsoft SQL Server 2014 and 2016 are also supported. This database is outside the boundary of the TOE and is only used for the storage of data. All data that is sent to the database is encrypted by the TOE and is stored in the database as cipherstrings. Decryption of data happens on the TOE after the data is retrieved from the database.

1.3.2 Security Functions provided by the TOE

The TOE provides the security functionality required by [SWAPP].

1.3.2.1 Cryptographic Support

The TOE relies on underlying cryptographic functionality provided by the platform for all of its cryptographic operations, as allowed by the [SWAPP].

1.3.2.2 Secure Software Update

The TOE is distributed as a .MSI installer package.

1.3.2.3 Security Management

The TOE does not come with any default credentials. Upon installation it will randomly generate a self-signed certificate, and AES 256 symmetric key and a GUID for the base configuration of the system. No data is stored by the application on the platform file system.

1.3.2.4 User Data Protection

The TOE does not store or transmit anything that could be considered Personally Identifiable Information (PII).

1.3.2.5 Protection of the TSF

The TOE employs several mechanisms to ensure that it is secure on the host platform. The TOE never allocates memory with both write and execute permission. The TOE is designed to operate in an environment in which the following security techniques are in effect:

- Data execution prevention,
- Mandatory address space layout randomization (no memory map to an explicit address),
- Structured exception handler overwrite protection,
- Export address table access filtering, and
- Anti-Return Oriented Programming.

This allows the TOE to operate in an environment in which the Enhanced Mitigation Experience Toolkit is also running. During compilation, the TOE is built with several flags enabled that check for engineering flaws. The TOE is built with the /GS flag enabled. This reduces the possibilities of stack-based buffer overflows in the product.

1.3.2.6 Trusted Path/Channels

TLS and SSH are used to protect all data transmitted to and from the TOE.

1.3.2.7 Unevaluated Functionality

The following functionality is outside the scope of the evaluation:

- Securing and protecting keys and certificates
- Providing visibility, threat intelligence, policy enforcement, and incident response for certificate-related outages and key compromises
- Integration with Venafi products and third-party applications – the evaluation is limited to secure communication channels
- Visibility into their key and certificate inventory, certificate reputation
- Issuance and renewal of certificates
- Policy enforcement
- Workflows
- Remediation of key and certificate misuse

1.3.3 Other References

Protection Profile for Application Software, version 1.3, dated 01 March 2019 [SWAPP].

Extended Package for Secure Shell, version 1.0, dated 19 February 2016 [SSHEP].

2 Conformance Claims

2.1 CC Conformance

This TOE is conformant to:

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

2.2 Protection Profile Conformance

This TOE is conformant to:

- Protection Profile for Application Software, version 1.3, dated 01 March 2019 [SWAPP].
- Extended Package for Secure Shell, version 1.0, dated 19 February 2016 [SSHEP].

2.3 Conformance Rationale

This Security Target provides exact conformance to Version 1.3 of the Protection Profile for Application Software and Version 1.0 of the Extended Package for Secure Shell (SSH). The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profile and the Extended Package, performing only operations defined there.

2.3.1 Technical Decisions

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

Identifier	Applicable	Exclusion Rationale (if applicable)
0495 – FIA_X509_EXT.1.2 Test Clarification	Yes	
0471 - Changes to App PP v1.3 related to PP-Modules	Yes	
0465 – Configuration Storage for .NET Apps	Yes	
0446 – Missing selections for SSH	Yes	
0445 – User Modifiable File Definition	Yes	
0444 – IPsec selections	Yes	
0435 – Alternative to SELinux for FPT_AEX_EXT.1.3	No	TOE does not run in Linux environment.
0437 – Supported Configuration Mechanism	Yes	
0434 – Windows Desktop Applications Test	Yes	
0427 – Reliable Time Source	Yes	
0416 – Correction to FCS_RBG_EXT.1 Test Activity	Yes	
0420 – Conflict in FCS_SSHC_EXT.1.1 and FCS_SSHS_EXT.1.1	Yes	
0332 – Support for RSA SHA2 host keys	Yes	
0331 – SSH Rekey Testing	Yes	
0240 – FCS_COP.1.1(1) Platform provided crypto for encryption/decryption	Yes	

3 Security Problem Definition

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

3.1 Threats

The following threats are drawn directly from the SWAPP.

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

Table 2 Threats

3.2 Assumptions

The following assumptions are drawn directly from the SWAPP.

ID	Assumption
A.PLATFORM[1]	The TOE relies upon a trustworthy computing platform with a reliable time clock for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.

[1] ST Application Note: This Assumption has been updated according to TD0427.

A.PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy.
A.PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software in compliance of the applied enterprise security policy.

Table 3 OSPs

3.3 Organizational Security Policies

There are no OSPs for the application

4 Security Objectives

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

4.1 Security Objectives for the TOE

The following security objectives for the TOE were drawn directly from the SWAPP.

ID	TOE Objective
O.INTEGRITY	Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom if ever shipped without errors, and the ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options. Addressed by: FDP_DEC_EXT.1, FMT_CFG_EXT.1, FPT_AEX_EXT.1, FPT_TUD_EXT.1
O.QUALITY	To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs. Addressed by: FMT_MEC_EXT.1, FPT_API_EXT.1, FPT_LIB_EXT.1, FPT_TUD_EXT.2, FCS_CKM.1(1)
O.MANAGEMENT	To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII. Addressed by: FMT_SMF.1, FPT_IDV_EXT.1, FPT_TUD_EXT.1, FPR_ANO_EXT.1
O.PROTECTED_STORAGE	To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data.

	Addressed by: FDP_DAR_EXT.1, FCS_STO_EXT.1, FCS_RBG_EXT.1, FCS_COP.1(1)
O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application. Addressed by: FTP_DIT_EXT.1, FCS_RBG_EXT.1, FCS_CKM_EXT.1, FCS_CKM.2, FDP_NET_EXT.1, FIA_X509_EXT.1

Table 4 Objectives for the TOE

4.2 Security Objectives for the Operational Environment

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

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

Table 5 Objectives for the environment

5 Security Requirements

This section identifies the Security Functional Requirements for the TOE and/or Platform. The Security Functional Requirements included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, dated: April 2017 and all international interpretations.

Requirement	Description
Mandatory SFRs	
FCS_CKM_EXT.1	Cryptographic Key Generation Services
FCS_COP.1(1)	Cryptographic Operation - Encryption/Decryption
FCS_RBG_EXT.1	Random Bit Generation Services
FCS_SSH_EXT.1	SSH Protocol
FCS_STO_EXT.1	Storage of Credentials
FDP_DEC_EXT.1	Access to Platform Resources
FDP_NET_EXT.1	Network Communications
FDP_DAR_EXT.1	Encryption of Sensitive Application Data
FMT_MEC_EXT.1	Supported Configuration Mechanism
FMT_CFG_EXT.1	Secure by Default Configuration
FMT_SMF.1	Specification of Management Functions
FPR_ANO_EXT.1	User Consent for Transmission of Personally Identifiable Information
FPT_API_EXT.1	Use of Supported Services and APIs

FPT_AEX_EXT.1	Anti-Exploitation Capabilities
FPT_TUD_EXT.1	Integrity for Installation and Update
FPT_LIB_EXT.1	Use of Third Party Libraries
FPT_IDV_EXT.1	Software Identification and Versions
FTP_DIT_EXT.1	Protection of Data in Transit
Optional, Selection-Based and Objective SFRs	
FCS_CKM.1(1)	Cryptographic Asymmetric Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_SSHC_EXT.1	SSH Protocol - Client
FIA_X509_EXT.1	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FPT_TUD_EXT.2	Integrity for Installation and Update

Table 6 SFRs

5.1 Conventions

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

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).
- Where operations were completed in the PP itself, the formatting used in the PP has been retained.

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

5.2 Security Functional requirements

5.2.1 Cryptographic Support (FCS)

FCS_CKM_EXT.1 Cryptographic Key Generation Services

FCS_CKM_EXT.1.1

The application shall [*invoke platform-provided functionality for asymmetric key generation*].

FCS_CKM.1(1) Cryptographic Asymmetric Key Generation

FCS_CKM.1.1(1)

The application shall [*invoke platform-provided functionality*] to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [

- ***[RSA schemes]*** *using cryptographic key sizes of* ***[2048-bit, 3072-bit]*** *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]***

].

FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1

The application shall [invoke platform-provided 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"],***

].

FCS_COP.1(1) Cryptographic Operation – Encryption/Decryption (Refined)

FCS_COP.1.1(1)

The SSH software shall [invoke platform-provided] encryption/decryption services for data in accordance with a specified cryptographic algorithm AES-CTR (as defined in NIST SP 800-38A) mode and cryptographic key sizes [128-bit, 256-bit].

ST Application Note
FCS_COP.1(1) was updated based on TD0240.

FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1

The application shall [use no DRBG functionality] for its cryptographic operations

FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1

The SSH software shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254 and [*5656, 6668*] as a [*client*].

FCS_SSHC_EXT.1 SSH Protocol – Client

FCS_SSHC_EXT.1.1

The SSH client shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, and [password-based].

FCS_SSHC_EXT.1.2

The SSH client shall ensure that, as described in RFC 4253, packets greater than [*35,000*] bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.3

The SSH software shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-ctr, aes256-ctr, [aes128-cbc, aes256-cbc].

FCS_SSHC_EXT.1.4

The SSH client shall ensure that the SSH transport implementation uses [*ssh-rsa, ecdsa-sha2-nistp256*] and [*no other public key algorithms*] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHC_EXT.1.5

The SSH client shall ensure that the SSH transport implementation uses [*hmac-sha1, hmac-sha2-256, hmac-sha2-512*] and [*no other MAC algorithms*] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.6

The SSH client shall ensure that [*diffie-hellman- group14-sha1*] and [*no other methods*] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.7

The SSH server shall ensure that the SSH connection be rekeyed after [*no more than 1 Gigabyte of data has been transmitted*] using that key.

FCS_SSHC_EXT.1.8

The SSH client shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key or [*no other methods*] as described in RFC 4251 section 4.1.

ST Application Note
FCS_SSHC_EXT.1 was updated based on TD0332 and TD0446.

FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1

The application shall [*invoke the functionality provided by the platform to securely store* *[DSN, PKCS12,* *PKCS8 (private key), Usernames, Passwords, Customer Application Credentials]*] to non-volatile memory.

5.2.2 User Data Protection (FDP)

FDP_DEC_EXT.1 Access to Platform Resources

FDP_DEC_EXT.1.1

The application shall restrict its access to [*network connectivity*].

FDP_DEC_EXT.1.2

The application shall restrict its access to [*system logs*].

FDP_NET_EXT.1 Network Communications

FDP_NET_EXT.1.1

The application shall restrict network communication to [*communications with the backend database, IIS application communication, communicating with managed hosts, communicating with external CA server, communicating with HSM, user configured discovery*]

FDP_DAR_EXT.1 Encryption Of Sensitive Application Data

FDP_DAR_EXT.1.1

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

5.2.3 Identification and Authentication (FIA)

FIA_X509_EXT.1 Certificate Validation

FIA_X509_EXT.1.1

The application shall [*invoked platform-provided 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*].
- The application shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - 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.
    - 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.

- Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the extendedKeyUsage field.

FIA_X509_EXT.1.2

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

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1

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

5.2.4 Security Management (FMT)

FMT_MEC_EXT.1 Supported Configuration Mechanism

FMT_MEC_EXT.1.1

The application shall [

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

ST Application Note
FMT_MEC_EXT.1 was updated based on TD0437.

FMT_CFG_EXT.1 Secure by Default Configuration

FMT_CFG_EXT.1.1

The application shall only provide 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 binaries and data files from modification by normal unprivileged users.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1

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

- enable/disable the transmission of any information describing the system's hardware, software, or configuration,
- enable/disable transmission of any application state (e.g. crashdump) information,
- *enable/disable debug level logging, enable/disable service modules, enable/disable web applications]*

].

5.2.5 Privacy (FPR)

FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information

FPR_ANO_EXT.1

The application shall [not transmit PII over a network].

5.2.6 Protection of TSF (FPT)

FPT_API_EXT.1 Use of Supported Services and APIs

FPT_API_EXT.1.1

The application shall only use supported platform APIs.

FPT_AEX_EXT.1 Anti-Exploitation Capabilities

FPT_AEX_EXT.1.1

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

FPT_AEX_EXT.1.2

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

FPT_AEX_EXT.1.3

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

FPT_AEX_EXT.1.4

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

FPT_AEX_EXT.1.5

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

FPT_TUD_EXT.1 Integrity for Installation and Update

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

FPT_TUD_EXT.1.3

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

FPT_TUD_EXT.1.4

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

FPT_TUD_EXT.1.5

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

FPT_TUD_EXT.2 Integrity for Installation and Update

FPT_TUD_EXT.2.1

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

FPT_TUD_EXT.2.2

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

FPT_LIB_EXT.1 Use of Third Party Libraries

FPT_LIB_EXT.1.1

- The application shall be packaged with only [*ISON.Net, PDFSharp, PFDSharp Charting, MigraDoc, MigraDoc.Rendering, MigraDoc.RTFRendering, HTMLAgility Pack, MS Anti-Cross Site Scripting Library, IronPython, jQuery, Moment JS, Backbone JS, twitter bootstrap Apache v2, underscore, Boost, Beast, JSON11, Base64, cxxopts, Chaos.NaCl*].

FPT_IDV_EXT.1 Software Identification and Versions

FPT_IDV_EXT.1.1

The application shall be versioned with [[*other version information*]]

5.2.7 Trusted Path/Channel (FTP)

FTP_DIT_EXT.1 Protection of Data in Transit

FTP_DIT_EXT.1.1

The application shall [

- *encrypt all transmitted* [*data*] with [*SSH as conforming to the Extended Package for Secure Shell*],
- *invoke platform-provided functionality to encrypt all transmitted data with* [*TLS*]

] between itself and another trusted IT product.

5.3 TOE SFR Dependencies Rationale for SFRs

The Protection Profile for Application Software contains all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the PP has been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the Protection Profile for Application Software which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the table below.

Assurance Class	Components	Components Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Life Cycle Support	ALC_CMC.1	Labeling of the TOE

Assurance Class	Components	Components Description
	ALC_CMS.1	TOE CM Coverage
	ALC_TSU_EXT.1	Timely Security Updates
Tests	ATE_IND.1	Independent Testing – Conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

Table 7 Security Assurance Requirements

5.5 Rationale for Security Assurance Requirements

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

5.6 Assurance Measures

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

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

SAR Component	How the SAR will be met
ALC_TSU_EXT.1	Venafi uses a systematic method for identifying and providing security relevant updates to the TOEs users via its support infrastructure. Users can report issues using the Venafi Customer Portal https://customerportal.venafi.com/ or by emailing support@venafi.com
ATE_IND.1	Venafi will provide the TOE for testing.
AVA_VAN.1	Venafi will provide the TOE for testing.

Table 8 TOE Security Assurance Measures

6 TOE Summary Specification

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

TOE SFR	Rationale
FCS_RBG_EXT.1	Due to its leveraging of platform cryptographic functionality there are no TOE functions covered by ST SFRs that use random numbers provided by the platform. All random numbers used by SFR related functions are used by the platform's underlying cryptographic functionality.
FCS_STO_EXT.1	The TOE relies on the platform to securely store the following: • DSN key • PKCS12 key • PKCS8 (private key) • Usernames • Passwords • Customer application credentials The Windows Registry is used for storage of the TOE's symmetric key. An AES 256 key is used for the encryption and decryption of secrets. It is protected by the Windows Data Protection API (DPAPI).
FCS_SSH_EXT.1 FCS_SSHC_EXT.1	The TOE functions as an SSH client in order to communicate with target applications and certificate authorities. Both public-key and password-based authentication are supported. The following SSH transport algorithms may be used: • AES128-CBC • AES256-CBC • AES128-CTR • AES256-CTR SSH-RSA and ECDSA-SHA2-NISTp256 are the supported public key algorithms. HMAC-SHA1, HMAC-SHA2-256 and HMAC-SHA2-512 may be used for data integrity. Diffiehellman-group14-sha1 is the only key exchange method used. If the TOE receives an SSH packet larger than 35,000 bytes the packet is dropped and the SSH connection is closed.
FDP_DEC_EXT.1 FDP_NET_EXT.1	Network connectivity is the only platform resource accessed by the TOE. The TOE communicates with a backend database, IIS applications, managed hosts, external CA servers, external HSMs and to perform discovery services. System logs are the only sensitive information repository accessed by the TOE. The TOE writes events to the system logs.
FDP_DAR_EXT.1	The only sensitive data stored by the TOE in non-volatile memory is listed in FCS_STO_EXT.1. No additional sensitive data is stored by the TOE.
FIA_X509_EXT.1 FIA_X509_EXT.2	The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for TLS connections. All certificate validation is performed by the underlying Windows platform, and certificates are stored in the Windows certificate store.

TOE SFR	Rationale
	As stated in the Windows ST, certificate validation is done in conformance to RFC 5280. Certificate validation paths must terminate with a trusted CA certificate that contains the basicConstraints extension and a CA flag that is set to TRUE. ExtendedkeyUsage field validation is also performed. CRLs are configurable and may be used for certificate revocation. Checking is also done for the basicConstraints extension and the cA flag to determine whether they are present and set to TRUE. If they are not, the certificate is not accepted. When the platform cannot establish a connection to a CRL distribution point to determine certificate validity the platform will reject the connection.
FMT_MEC_EXT.1	The TOE does not store any TSF related configuration options using platform mechanisms.
FMT_CFG_EXT.1	There are no default credentials within the TOE. Upon installation the TOE generates a GUID for the base configuration of the system. The TOE binaries and installer are signed and installed with permissions to ensure unprivileged users cannot modify the TOE binaries.
FMT_SMF.1	The TOE is capable of the following management functions: • Logging: It is possible to enable/disable Debug level logging. Debug level logging has the potential to display internal information. Debug level logging can be enabled/disabled via an option in the UI on the Engine (Platform Tree → Engine → Allow Debug) • Stack Traces: By default stack traces are not displayed in the Admin UI consoles. In order to enable the display of stack traces it is necessary to modify the web.config file for each application. • Enable/Disable Service Modules: It is possible to enable/disable functions of the platform. In order to do so, go to the Platforms Tree and Enable/Disable desired modules. • Web Applications: Upon install, the Admin is given the option to enable various web applications. Once created, these applications can be modified by running the Venafi Control Center.
FPR_ANO_EXT.1	The TOE does not transmit any PII.
FCS_CKM_EXT.1, FCS_CKM.1(1), FCS_CKM.2, FCS_COP.1(1), FPT_API_EXT.1	Microsoft .Net 4.6.1 is used by the TOE. Through .Net the TOE is able to call the following underlying Windows cryptographic modules: • Cryptographic Primitives Library (bcryptprimitives.dll and ncryptsslp.dll) (FIPS Approved algorithms: AES (Cert. #2832); DRBG (Certs. #489); DSA (Cert. #855); ECDSA (Cert. #505); HMAC (Cert. #1773); KAS (Cert. #47); KBKDF (Cert. #30); PBKDF (vendor affirmed); RSA (Certs. #1487, #1493 and #1519); SHS (Cert. #2373); Triple-DES (Cert. #1692)) • Kernel Mode Cryptographic Primitives Library (cng.sys) (FIPS Approved algorithms: AES (Cert. #2832); DRBG (Certs. #489); ECDSA (Cert. #505); HMAC (Cert. #1773); KAS (Cert. #47); KBKDF (Cert. #30); PBKDF (vendor affirmed); RSA (Certs. #1487, #1493 and #1519); SHS (Cert. # 2373)) The TOE uses the cryptographic modules for SSH and TLS connections. The TLS protocol is also provided by .Net.

TOE SFR	Rationale
	The key establishment schemes are RSA-based with key sizes of 2048- or 3072-bits or elliptic curve-based with NIST curves, P-256, P-384, or P-521. Both of these schemes are used for TLS. The scheme used is dependent on the selected cipher suites. For SSH, the key establishment schemes are elliptic curve-based with NIST curve P-256 and SSH-RSA.
FPT_AEX_EXT.1	The TOE never maps memory to explicit addresses, nor does it allocate memory regions with write and execute permissions. It is not necessary to use compiler flags to enable ASLR. The TOE's code is not run natively, but instead as managed code on top of Microsoft's .Net. Similarly, the use of a managed code base means that compiler flags aren't used for stack-based buffer overflow protection. Stack Based buffer overflows are protected in managed code by an exception being thrown by the CLR rather than having the overflow happen on the stack.
FPT_TUD_EXT.1, FPT_TUD_EXT.2	Updates to the TOE are distributed as .MSI installation files and are performed in the same manner as a product installation. All binaries are signed using signtool.exe, which is a .Net framework tool for digital file signatures. TOE version can be found in the WebAdmin UI in the 'About Trust Protection Platform' selection or on the REST API with the 'SystemStatus/Version' call. The removal of the TOE installation package results in the deletion of all traces of the application.
FPT_LIB_EXT.1	The TOE is installed with the following third-party libraries: • JSON.Net • PDFSharp • PFDSharp Charting • MigraDoc • MigraDoc.Rendering • MigraDoc.RTFRendering • HTMLAgility Pack • MS Anti-Cross Site Scripting Library • IronPython • jQuery • Moment JS • Backbone JS • twitter bootstrap Apache v2 • underscore • Boost • Beast • JSON11 • Base64 • cxxopts • Chaos.NaCl
FPT_IDV_EXT.1	The application version follows a major.minor.patch.build structure. The build corresponds to a git tag for that particular build.
FTP_DIT_EXT.1	All external communications are protected by SSH or TLS. The TLS protocol is provided by

TOE SFR	Rationale
	the underlying platform. The TOE uses .NET to invoke the platform’s TLS functionality.

Table 9 TOE Summary Specification SFR Description