Venafi Trust Protection Platform v23.1

Security Target

Document Version: 1.6

Table Of Contents

Table Of Tables

Revision History

Version	Date	Description
0.0	September 2022	Initial release
0.1	February 2023	Cleaned up formatting and aligned with gap analysis results.
0.2	May 2023	Updates after meeting with vendor.
0.3	August 2023	Removed Curve448-sha512; update FCS_RBG_EXT.1; added FCS_SSHC_EXT.1 to Security Functional Requirements (was already in TSS).
0.4	September 2023	Removed json2.js from third party libraries; removed ssh-ed448 (RFC 8709) from FCS_SSHC_1.1.
0.5	September 2023	Fixed sync issues between various third party lists; removed OCSP.
0.6	December 2023	Addressed ECR comments including updating title version number; removing all archived TDs; added TOE Environment section. Also updated SQL server version.
0.7	January 2024	Updates addressing ECR, AAR, and ETR comments.
1.0	January 2024	Preparation of final version for checkout.
1.1	February 2024	Updates to section 1.3.1 and 1.3.2
1.2	February 2024	Update to section 1.3.1 minor addition to final sentence including all devices of figure 1. Final version for checkout.
1.3	March 2024	Updated FMT_MEC.1 in TSS.
1.4	May 2024	Added TDs, added section 7 CAVP Algorithm Certificate Details.
1.5	June 2024	Minor updates based on final pre-CO review.
1.6	July 2024	Based on ECR comments

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 v23.1 Security Target
ST Version	1.6
ST Date	12 July, 2024
ST Author	Acumen Security, LLC.
TOE Identifier	Venafi Trust Protection Platform
TOE Software Version	23.1
TOE Developer	Venafi
Key Words	Software

Table 1 TOE/ST Identification

1.2 TOE Overview and Description

Venafi Trust Protection Platform is a windows application that secures and protects keys and certificates. 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 1.3.3.1 through 1.3.3.6 for an identification of the evaluated functionality and section 1.3.3.7 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. The TOE is a Windows Application. For this evaluation the TOE runs on Windows Server 2016 Standard configured in FIPS mode running on a server with an Intel Xeon processor with AES-NI and PCLMULQDQ and SSSE 3. 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.

- IronPython
- Chaos.NaCl
- Microsoft Intune CSR Validation
- Sustainsys Saml2
- Excelsior JET
- F5 iControl Assembly for .NET
- Bootstrap
- Backbone
- Underscore
- Jquery
- date.js
- dateRangePicker.js, dateRangePicker.css
- moment.js
- easyDate.js
- maskedInput.js
- browser.js
- jquery.timepicker.js
- Select2.js
- moment-timezone.js
- core.js
- dropzone.js
- JSON.Net
- ASP.NET Web Stack
- Sencha Ext JS
- Tigra Calendar
- Pretty-Print JSON
- D3.js
- chart.js
- mustache.js

The TOE provides three consoles for management:

- A web-based console that can be launched by connecting to the TOE using a browser.
- Venafi Configuration Console (VCC): A powerful Microsoft Management Console (MMC) is a snap-in console that allows an administrator to manage Venafi services, enable product components, configure database settings.

- WinAdmin: A Windows-based console that runs locally on the Trust Protection Platform server.

The TOE also uses an external database to store credentials, certificates, keys and log data. Microsoft SQL Server 2022 Developer is used in the evaluated configuration and Microsoft SQL Server 2014, 2016 SP2, 2017, and 2019 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. The TOE supports local as well as a remote database.

The TOE provides following connections:

- The TOE leverages Microsoft's IIS to provide web services for User or Admin authentication to access the web-based console
- The TOE connects to a remote database securely over TLS
- The TOE acts as a client and connects securely with managed hosts over SSH
- The TOE acts as a client and connects securely over TLS to perform discovery services
- The TOE communicates with a CA server over HTTP to validate the presented server's certificate by retrieving CRLs

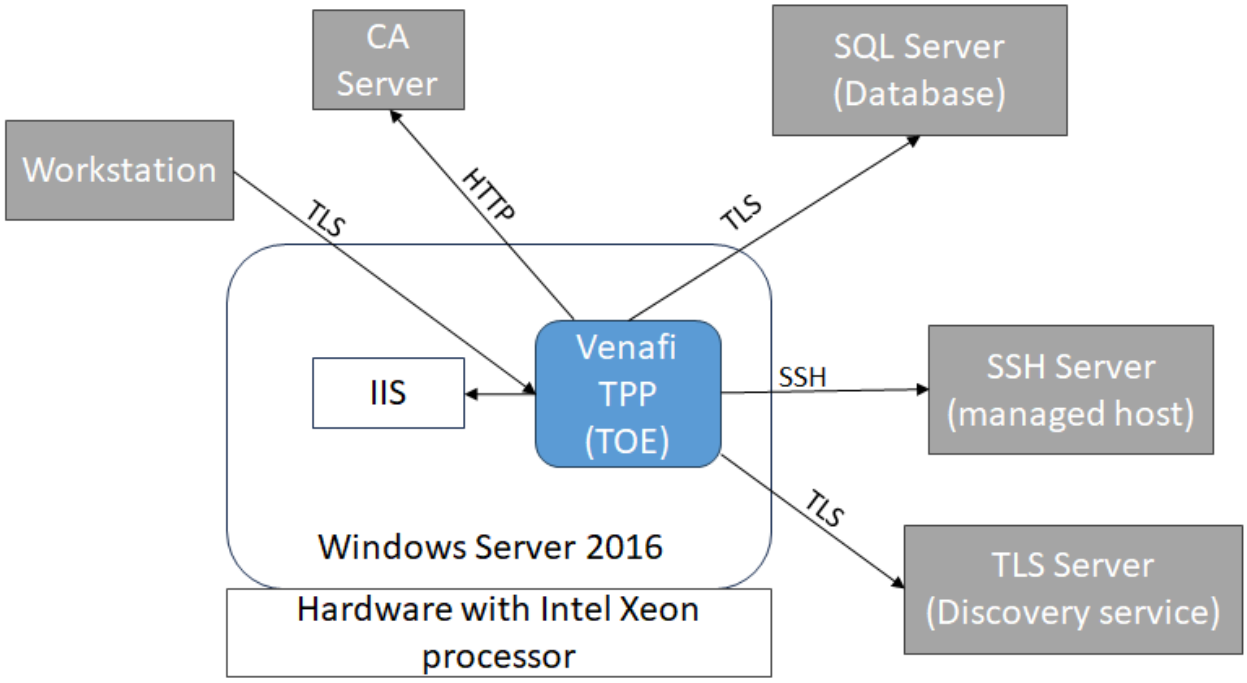

Figure 1 TOE network diagram

1.3.2 TOE Environment

The following components must be present in the operational environment to operate the TOE in the evaluated configuration:

Table 2 Operational Environment Components

Component	Required	Purpose/Description
Workstation	Optional	Workstation to access the TOE via web-based console over TLS.
SQL Server (Database)	Yes	The TOE uses an external database to store credentials, certificates, keys and log data. Microsoft SQL Server 2022 Developer is used in the evaluated configuration. The connection to a remote database is secured over TLS.
TLS Server (Discovery Service)	Optional	This is a IP-based target machine on which discovery services can be performed to discover the SSL certificates. The TOE communicates securely over TLS.
CA Server	Optional	This is a CRL server that provides a list of certificates that have been revoked. It is used by the TOE to check a server's presented certificate revocation status. The TOE communicates to the CA/CRL server over HTTP.
SSH Server (managed host)	Yes	This is a remote system managed host. TOE connects to the managed host over SSH

1.3.3 Security Functions provided by the TOE

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

1.3.3.1 Cryptographic Support

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

1.3.3.2 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.

1.3.3.3 Privacy

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

1.3.3.4 User Data Protection

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).

No additional sensitive data is stored by the TOE.

1.3.3.5 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.

1.3.3.6 Trusted Path/Channels

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

1.3.3.7 Unevaluated Functionality

The following functionality is outside the scope of the evaluation:

- 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

1.3.4 Other References

Protection Profile for Application Software, version 1.4, dated 22 April 2022 [SWAPP].

Functional Package for Secure Shell, version 1.0, dated 13 May 2021 [SSHFP].

2 Conformance Claims

2.1 CC Conformance

The 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 ST and TOE claims exact conformance to:

- Protection Profile for Application Software, version 1.4, dated 07 October 2021 [SWAPP].
- Functional Package for Secure Shell, version 1.0, dated 13 May 2021 [SSHFP].

2.3 Conformance Rationale

This Security Target provides exact conformance to Version 1.4 of the Protection Profile for Application Software and Version 1.0 of the Functional 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 Functional Package, performing only operations defined there.

The TOE type is application software.

2.3.1 Technical Decisions

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

Table 3 - SWAPP TDs

Identifier	Applicable	Exclusion Rationale (if applicable)
0823 - Update to Microsoft Windows Exploit Protection link in FPT_AEX_EXT.1.3	Yes	
0822 - Correction to Windows Manifest File for FDP_DEC_EXT.1	Yes	
0815 - Addition of Conditional TSS Activity for FPT_AEX_EXT.1.5	Yes	
0798 – Static Memory Mapping Exceptions	Yes	
0780 – FIA_X509_EXT.1 Test 4 Clarification	Yes	
0756 – Update for platform-provided full disk encryption	Yes	
0747 – Configuration Storage Option for Android	No	TOE is not an Android device.
0743 – FTP_DIT_EXT.1.1 Selection exclusivity	Yes	
0736 – Number of elements for iterations of FCS_HTTPS_EXT.1	No	SFR not claimed for evaluation.

Identifier	Applicable	Exclusion Rationale (if applicable)
0719 – ECD for PP APP V1.3 and 1.4	Yes	
0717 – Format changes for PP_APP_V1.4	Yes	
0664 – Testing activity for FPT_TUD_EXT.2.2	Yes	
0650 – Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4	No	TOE is not a VPN client.
0628 – Addition of Container Image to Package Format	Yes	

Table 4 – SSHFP TDs

Identifier	Applicable	Exclusion Rationale (if applicable)
0777 – Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	No	App PP does not include FAU_GEN.1.
0732 – FCS_SSHS_EXT.1.3 Test 2 Update	Yes	
0695 – Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package.	Yes	
0682 – Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	No	SSH Server is not claimed.

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 (OSPs) 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 5 Threats

3.2 Assumptions

The following assumptions are drawn directly from the SWAPP.

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

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

Table 6 OSPs

3.3 Organizational Security Policies

The PP and Functional Package do not define any OSPs.

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_EXT.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 7 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 8 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_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/AK	Cryptographic Asymmetric Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_COP.1/SKC	Cryptographic Operation - Encryption/Decryption
FCS_COP.1/Hash	Cryptographic Operation - Hashing
FCS_COP.1/KeyedHash	Cryptographic Operation - Keyed-Hash Message Authentication
FCS_COP.1/Sig	Cryptographic Operation - Signing
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 9 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 *italicized 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/AK Cryptographic Asymmetric Key Generation

FCS_CKM.1.1/AK

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 or greater]*** *that meet the following:* ***[FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3]****,*
- ***[ECC schemes]*** *using* ***["NIST curves" P-384 and [P-256, P-521]]*** *that meet the following:* ***[FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4]*** *,*
- *[****FFC Schemes****] **using ["safe-prime" groups]** that meet the following:* ***[NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"*** *and [RFC 3526]****].***

].

Note: TD0717 applied.

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 meet the following:* ***RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1"*** *,*
- ***[Elliptic curve-based key establishment schemes]*** *that meets the following:* ***[NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"],***
- ***[FFC Schemes using "safe-prime" groups]*** *that meet the following:* ***'NIST Special Publication***

800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" *and [RFC 3526].*

].

FCS_COP.1/SKC Cryptographic Operation - Encryption/Decryption

FCS_COP.1.1/SKC

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

- *AES-CBC (as defined in NIST SP 800-38A) mode,*
- *AES-CTR (as defined in NIST SP 800-38A) mode*

] and cryptographic key sizes [*128-bit, 256-bit*].

TD0717 applied.

FCS_COP.1/Hash Cryptographic Operation - Hashing

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

- *SHA-1,*
- *SHA-256,*
- *SHA-384,*
- *SHA-512,*

] and message digest sizes [

- *160*
- *256,*
- *384,*
- *512,*

] bits that meet the following: [FIPS Pub 180-4].
TD0717 applied.

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

FCS_COP.1.1/KeyedHash

The **application** shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm [

- *HMAC-SHA-256*
- *HMAC-SHA-384*
- *HMAC-SHA-512*

] and [

- *no other algorithms*

] with key sizes [*256, 384, 512*] and message digest sizes [*256, 384, 512]* and [*no other size*] bits that meet the following: [FIPS Pub 198-1, '*The Keyed-Hash Message Authentication Code'* and FIPS Pub 180-4 '*Secure Hash Standard'*].
TD0717 applied.

FCS_COP.1/Sig Cryptographic Operation - Signing

FCS_COP.1.1/Sig

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

- ***RSA schemes** using cryptographic key sizes of [2048-bit or greater] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5],*
- ***ECDSA schemes** using ["NIST curves" P-256, P-384 and [**P-521**]] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6]*

].
TD0717 applied.

FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1

The application shall [*invoke platform-provided DRBG functionality*] for its cryptographic operations.

FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1

The TOE shall implement SSH acting as a [client] in accordance with that complies with RFCs 4251, 4252, 4253, 4254 and [*4344, 5656, 6668, 8268, 8308, 8332, 8709, 8731*] and [no other standard].

(Note: The following mapping is provided as a guide to ST authors to ensure the appropriate RFC selections are made based on applicable selections in subsequent SFRs:

- RFC 4256: Select for keyboard-interactive authentication
- RFC *4344*: Select for AES-128-CTR or AES-256-CTR
- RFC 5647: Select for AEAD_AES_128_GCM, AEAD_AES_256_GCM, or aes*-gcm@openssh.com
- RFC *5656*: Select for elliptic curve cryptography
- RFC 6187: Select for X.509 certificate use
- RFC *6668*: Select for HMAC-SHA-2 algorithms

- RFC *8268*: Select for FFC DH groups with SHA-2
- RFC *8308*: Select if RFC 8332 is selected
- RFC *8332*: Select if SHA-2 is available with ssh-rsa
- RFC *8709*: Select if ed25519 or ed448 is used as a public key algorithm
- RFC *8731*: Select if curve25519 or curve448 is used for key exchange

).

FCS_SSH_EXT.1.2

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [

- *"password" (RFC 4252),*
- *"publickey" (RFC 4252): [*
    - *ssh-rsa (RFC 4253),*
    - *rsa-sha2-256 (RFC 8332),*
    - *rsa-sha2-512 (RFC 8332),*
    - *ecdsa-sha2-nistp256 (RFC 5656),*
    - *ecdsa-sha2-nistp384 (RFC 5656),*
    - *ecdsa-sha2-nistp521 (RFC 5656),*
    - *ssh-ed25519 (RFC 8709),*

    ]

] and no other methods.

FCS_SSH_EXT.1.3

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

FCS_SSH_EXT.1.4

The TSF shall protect data in transit from unauthorized disclosure using the following mechanisms: [

- *aes128-ctr (RFC 4344),*
- *aes256-ctr (RFC 4344),*
- *aes128-cbc (RFC 4253),*
- *aes256-cbc (RFC 4253),*

] and no other mechanisms.

FCS_SSH_EXT.1.5

The TSF shall protect data in transit from modification, deletion, and insertion using: [

- *hmac-sha2-256 (RFC 6668),*

- *hmac-sha2-512 (RFC 6668),*

] and no other mechanisms.

FCS_SSH_EXT.1.6

The TSF shall establish a shared secret with its peer using: [

- *diffie-hellman-group14-sha256 (RFC 8268),*
- *diffie-hellman-group16-sha512 (RFC 8268),*
- *diffie-hellman-group18-sha512 (RFC 8268),*
- *ecdh-sha2-nistp256 (RFC 5656),*
- *ecdh-sha2-nistp384 (RFC 5656),*
- *ecdh-sha2-nistp521 (RFC 5656),*
- *curve25519-sha256 (RFC 8731),*

] and no other mechanisms.

FCS_SSH_EXT.1.7

The TSF shall use SSH KDF as defined in [

- *RFC 4253 (Section 7.2),*
- *RFC 5656 (Section 4)*

] to derive the following cryptographic keys from a shared secret: *session keys*.

FCS_SSH_EXT.1.8

The TSF shall ensure that [

- *a rekey of the session keys,*

] occurs when any of the following thresholds are met:

- one hour connection time
- no more than one gigabyte of transmitted data, or
- no more than one gigabyte of received data.

FCS_SSHC_EXT.1 SSH Protocol – Client

FCS_SSHC_1.1

The TSF shall authenticate its peer (SSH server) using: [

- *using a local database by associating each host name with a public key corresponding to the following list: [*
  - *ssh-rsa (RFC 4253),*
  - *rsa-sha2-256 (RFC 8332),*

- *rsa-sha2-512 (RFC 8332),*
- *ecdsa-sha2-nistp256 (RFC 5656),*
- *ecdsa-sha2-nistp384 (RFC 5656),*
- *ecdsa-sha2-nistp521 (RFC 5656),*
- *ssh-ed25519 (RFC 8709),*

]

] as described in RFC 4251 section 4.1.

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

- *user-initiated communication for [User/Admin authentication over web-based console, user configured discovery]*
- [*communications with the backend database, communicating with managed hosts, communicating with CA servers*]

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 [*invoke 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, and that any path constraints

are met.

- The application shall validate that any CA certificate includes caSigning purpose in the key usage field
- The application shall validate the revocation status of the certificate using [*CRL as specified in RFC 5280 Section 6.3, CRL as specified in RFC 8603*].
- The application shall validate the extendedKeyUsage (EKU) field according to the following rules:
    - 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 EKU 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 EKU 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 EKU 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 EKU 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 EKU 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 EKU 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.

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].

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 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 documented platform APIs.

FPT_AEX_EXT.1 Anti-Exploitation Capabilities

FPT_AEX_EXT.1.1

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

Note: TD0798 applied.

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 built 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

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

FPT_TUD_EXT.1.5

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

FPT_TUD_EXT.2 Integrity for Installation and Update

FPT_TUD_EXT.2.1

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

Note: TD0628 applied.

FPT_TUD_EXT.2.2

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

FPT_TUD_EXT.2.3

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

FPT_LIB_EXT.1 Use of Third Party Libraries

FPT_LIB_EXT.1.1

The application shall be packaged with only [The TOE is installed with the following third-party libraries:

- IronPython
- Chaos.NaCl
- Microsoft Intune CSR Validation
- Sustainsys Saml2
- Excelsior JET
- F5 iControl Assembly for .NET
- Bootstrap
- Backbone
- Underscore
- Jquery
- date.js
- dateRangePicker.js, dateRangePicker.css
- moment.js
- easyDate.js
- maskedInput.js
- browser.js
- jquery.timepicker.js
- Select2.js
- moment-timezone.js
- core.js
- dropzone.js

- JSON.Net
- ASP.NET Web Stack
- Sencha Ext JS
- Tigra Calendar
- Pretty-Print JSON
- D3.js
- chart.js
- mustache.js

FPT_IDV_EXT.1 Software Identification and Versions

FPT_IDV_EXT.1.1

The application shall be versioned with [[*unique version numbering system*]].

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 defined in the Functional Package for Secure Shell for [connecting to remote systems (managed hosts)]],*
- *invoke platform-provided functionality to encrypt all transmitted data with* [*TLS] for [authentication, remote database, Discovery Services]*

] between itself and another trusted IT product.

Note: TD0743 applied.

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
	ALC_CMS.1	TOE CM Coverage
	ALC_TSU_EXT.1	Timely Security Updates

Assurance Class	Components	Components Description
Security Target Evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Stated problem definition
	ASE_TSS.1	TOE summary specification
Tests	ATE_IND.1	Independent Testing – Conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

Table 10 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.

SAR Component	How the SAR will be met
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 ALC_CMS.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_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 There are no third-party processes. Policies covering time between vulnerability disclosure and availability of remediation: Venafi does not have a policy currently in place specifically for response times regarding public disclosure of vulnerabilities, however Venafi does have an existing policy for vulnerability SLAs for the product: • Critical: 14 days • High: 60 days • Medium: By the time of next major patch release. (e.g. 24.1 -> 24.3) • Low: By the time of next major patch release. Secure delivery of TOE to customers: Venafi follows a standard process for developing, testing, building, and releasing the TOE per the Agile SAFe framework and OWASP SAMM v2 to build a secure product to provide to customers. All builder devices, developer machines, and download site hosting solutions receive security updates on a regular basis, and are managed and kept up to date following best practices established by NIST and CIS, following a standard company policy. As part of the software build process, hashes of the software are made and compared at several points, and the expected hashes are provided to customers allowing them to compare the hash of the downloaded content to the expected value.
ATE_IND.1	Venafi will provide the TOE for testing.
AVA_VAN.1	Venafi will provide the TOE for testing.

Table 11 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
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 There are no third-party processes. Policies covering time between vulnerability disclosure and availability of remediation: Venafi does not have a policy currently in place specifically for response times regarding public disclosure of vulnerabilities, however Venafi does have an existing policy for vulnerability SLAs for the product: • Critical: 14 days • High: 60 days • Medium: By the time of next major patch release. (e.g. 24.1 -> 24.3) • Low: By the time of next major patch release. Secure delivery of TOE to customers: Venafi follows a standard process for developing, testing, building, and releasing the TOE per the Agile SAFe framework and OWASP SAMM v2 to build a secure product to provide to customers. All builder devices, developer machines, and download site hosting solutions receive security updates on a regular basis, and are managed and kept up to date following best practices established by NIST and CIS, following a standard company policy. As part of the software build process, hashes of the software are made and compared at several points, and the expected hashes are provided to customers allowing them to compare the hash of the downloaded content to the expected value.
FCS_RBG_EXT.1	The TOE invokes platform provided DRBG (SecureRNG) for the purpose of generating a salt value, which is used to protect the Windows Data Protection API (DPAPI) key. The TOE invokes underlying platform's SecureRNG using the System.Security.Cryptography.RandomNumberGenerator class, which uses BCryptGenRandom. All random numbers used by the TLS and SSH SFR related functions are used by the platform's underlying cryptographic functionality indirectly.
FCS_STO_EXT.1	The TOE relies on the platform to securely store the following: • DSN key – related to database connection data – the primary use case is in installation answer files that refer to DNS keys to connect to the existing TOE database. • PKCS12 key - related to the product capability to export certificates and their private keys. This format is an option when certificate data is exported. • PKCS8 (private key) - related to the product capability to export certificates and their private keys. This format is an option when certificate data is exported. • Usernames, Passwords, and Customer application credentials are data types stored as hashes: ○ Application credentials are for certificate and key management of devices managed by the TOE – stored in a hashed format.

TOE SFR	Rationale
	○ Usernames and passwords are used both for credentials for access to the product, and for other products as application credentials. Best practices are followed in all cases and sensitive data is stored not in plain text and always as salted hashes. All certificates are stored in the Windows Certificate Store. 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). All usernames, passwords, and credentials are protected by the Windows 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. The TOE implements SSH acting as a client in accordance RFCs 4251, 4252, 4253, 4254, 4344, 5656, 6668, 8268, 8308, 8332, 8709, 8731. 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, RSA-SHA2-256, RSA-SHA2-512, ECDSA-SHA2-NISTp256, ECDSA-SHA2-NISTp384, ECDSA-SHA2-NISTp521 and SSH-ED25519 are the supported public key algorithms. HMAC-SHA2-256 and HMAC-SHA2-512 may be used for data integrity. The TOE supports following algorithms to establish shared secret with its peer: • *diffie-hellman-group14-sha256*, • *diffie-hellman-group16-sha512*, • *diffie-hellman-group18-sha512*, • *ecdh-sha2-nistp256*, • *ecdh-sha2-nistp384*, • *ecdh-sha2-nistp521*, • *curve25519-sha256*, The TOE supports SSH KDF as per RFC 4253 (Section 7.2) and RFC 5656 (Section 4). TOE examines the packet_length field to determine whether the packet is a large packet or not. If the TOE receives an SSH packet larger than 35,000 bytes the packet is dropped and the SSH connection is closed. The TOE enforces SSH session rekey if the connection time exceeds one hour or the data transfer (sent/received) exceed one gigabyte.
FDP_DEC_EXT.1 FDP_NET_EXT.1	Network connectivity is the only platform hardware resource accessed by the TOE. The TOE leverages Microsoft IIS webserver for User or Admin authentication over web-based console. The TOE also communicates with an external database, managed hosts, CA servers, 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	The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for

TOE SFR	Rationale
FIA_X509_EXT.2	TLS connections. All certificate validation is performed by invoking the underlying Windows platform, and certificates are stored in the Windows certificate store. The TOE supports a chain length four or greater. 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 application cannot establish a connection to a CRL distribution point to determine certificate validity the application will reject the connection. The TOE does not support EC certificates.
FMT_MEC_EXT.1	As a Windows application, the TOE utilizes both the Windows Registry and C:\ProgramData\ directory, where changes can be observed when configuration changes are made to the TOE. The TOE supports following configuration settings related to any SFRs and any settings that are mandated in the AGD: • all management functions covered under FMT_SMF.1 • database configuration • managed host configuration • discovery services configuration • certificate verification configuration • CRL check configuration
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. A master administrator role (admin) is created and the password for this account is defined as part of the installation. 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/AK,	Microsoft .Net 4.7.2 is used by the TOE, which is listed here: https://learn.microsoft.com/en-us/dotnet/api/?view=netframework-4.7.2

TOE SFR	Rationale
FCS_CKM.2, FPT_API_EXT.1	Through .Net the TOE is able to call the underlying Windows cryptographic modules. Please see section **7 CAVP Algorithm Certificate Details** for further information. The SSH and TLS components of TPP makes use of .NET cryptographic modules for the encryption and decryption of data. SSH itself is provided by Venafi's maintained internal branch of Maverick, while the TLS protocol is provided by Microsoft .NET alongside TLS cryptography. The key generation schemes are RSA-based with key sizes of 2048-bits or 3072-bits, or elliptic curve-based with NIST curves, P-256, P-384, or P-521, or FFC schemes using "safe-prime" groups. These schemes are used for both TLS and SSH. The key establishment schemes are RSA-based RSAES-PKCS1-v1_5, or elliptic curve-based with NIST curves, P-256, P-384, or P-521, or FFC schemes using "safe-prime" groups. All three 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, P-384, P-521, and FFC schemes using "safe-prime" groups.
FCS_COP.1/Hash FCS_COP.1/SKC, FCS_COP.1/Sig, FCS_COP.1/KeyedHash	The TOE uses the underlying platform .NET cryptographic modules for all hashing, keyed-hashing, encryption and digital signature generation and verification functions. The TOE uses AES-CBC and AES-CTR with 128-bit and 256-bit key sizes for encryption and decryption of data as part of SSH trusted channel. The TOE uses SHA-1, SHA-256, SHA-384, and SHA-512 for cryptographic hashing as part of SSH trusted channel. Hash functions are used as part of public key authentication, protecting data in transit (via HMAC), key exchange, and digital signatures as part of SSH trusted channel. The TOE uses HMAC (SHA-256, SHA-384, and SHA-512) for cryptographic keyed-hash message authentication as part of SSH trusted channel. The TOE uses ECDSA Signature Generation, and Signature Verification as part of SSH trusted channel. NIST curves P-256, P-384 and P-521 are supported. The TOE uses RSA Signature Generation and Signature Verification as part of SSH trusted channel. Key sizes of 2048 and 3072 are supported. Please see section **7 CAVP Algorithm Certificate Details** for further information.
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. This is done by default. 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	The application is distributed as an additional software package to the platform OS. 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 by Venafi using signtool.exe, which is a .Net framework tool for digital file signatures. Venafi is the only authorized source to sign the executable binary.

TOE SFR	Rationale
	Authorized source can be verified by right-clicking the .MSI file and select Properties. Under Digital Signatures Tab, Name of signer will indicate “Venafi, Inc”. Additionally, ensure that the binaries are downloaded from the authorized source – via https://download.venafi.com/. Users must have a username and password to login to download the binaries. TOE version can be found in the WebAdmin UI in the ‘About Trust Protection Platform’. The removal of the TOE installation package results in the deletion of all traces of the application. The product has functionality built in to inform administrators when a patch for the version in use is released, or when a new version is released. This functionality will not work if there is no external network access .
FPT_LIB_EXT.1	The TOE is installed with the following third-party libraries: • IronPython • Chaos.NaCl • Microsoft Intune CSR Validation • Sustainsys Saml2 • Excelsior JET • F5 iControl Assembly for .NET • Bootstrap • Backbone • Underscore • Jquery • date.js • dateRangePicker.js, dateRangePicker.css • moment.js • easyDate.js • maskedInput.js • browser.js • jquery.timepicker.js • Select2.js • moment-timezone.js • core.js • dropzone.js • JSON.Net • ASP.NET Web Stack • Sencha Ext JS • Tigra Calendar • Pretty-Print JSON • D3.js • chart.js • mustache.js
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 TOE provides following secure connections:

TOE SFR	Rationale
	• The TOE leverages Microsoft's IIS to provide web services for User or Admin authentication to access the web-based console over TLS • The TOE connects to a remote database securely over TLS • The TOE acts as a client and connects securely with managed hosts over SSH • The TOE acts as a client and connects securely over TLS to perform discovery services SSH protocol is provided by Venafi's maintained internal branch of Maverick, while the TLS protocol is provided by the underlying platform. The TOE uses .NET to invoke the platform's TLS functionality. The TOE uses System.Net APIs and System.ServiceModel APIs to leverage the underlying platform's TLS functionality. The platform provides support for the following TLS 1.2 cipher suites in evaluated configuration: • TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246, • TLS_RSA_WITH_AES_256_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_128_GCM_SHA256 as defined in RFC 5288, • TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288, • TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246, • TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246, • TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288 • TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 8422 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 8422 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 The TOE supports remote authentication over web-based console using username and password. Credentials are transmitted over TLS. The TOE also transmits credentials of the managed hosts over SSH.

Table 12 TOE Summary Specification SFR Description

7 CAVP Algorithm Certificate Details

Table 13 CAVP Algorithm Certificate Details

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
FCS_CKM.1/AK	[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.	Microsoft Windows 10 Anniversary Update, Windows Server 2016, Windows Storage Server 2016; Microsoft Surface Book, Surface Pro 4, Surface Pro 3 and Surface 3 w/ Windows 10 Anniversary Update; Microsoft Lumia 950 and Lumia 650 w/ Windows 10 Mobile Anniversary Update RSA Key Generation Implementation	RSA KeyGen (FIPS186-4)	RSA #2195
	[ECC schemes] using ["NIST curves" P-384 and [P-256, P-521]] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4].	Microsoft Windows 10 Anniversary Update, Windows Server 2016, Windows Storage Server 2016; Microsoft Surface Book, Surface Pro 4, Surface Pro 3 and Surface 3 w/ Windows 10 Anniversary Update; Microsoft Lumia 950 and Lumia 650 w/ Windows 10 Mobile Anniversary Update MsBignum Cryptographic Implementations	ECDSA KeyGen (FIPS186-4) ECDSA KeyVer (FIPS186-4)	ECDSA #911
	[FFC Schemes] using ["safe-prime" groups] that meet the following: [NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key	N/A	No NIST CAVP, CCTL has performed all assurance/evaluation activities and documented in the ETR and AAR accordingly.	N/A. Testing for FFC Schemes using safe-prime groups is done as part of

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
	Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526]].			testing in CKM.2.1.
FCS_CKM.2	[RSA-based key establishment schemes] that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1" ,	N/A	No NIST CAVP, CCTL has performed all assurance/evaluation activities and documented in the ETR and AAR accordingly.	N/A. This testing was performed in conjunction with FTP_DIT_EXT.1 to demonstrate correct operation.
	[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"].	Microsoft Windows 10 Anniversary Update, Windows Server 2016, Windows Storage Server 2016; Microsoft Surface Book, Surface Pro 4, Surface Pro 3 and Surface 3 w/ Windows 10 Anniversary Update; Microsoft Lumia 950 and Lumia 650 w/ Windows 10 Mobile Anniversary Update Cryptography Next Generation (CNG) Implementations	KAS-ECC	KAS #92
	[FFC Schemes using "safe-prime" groups] that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526].	N/A	No NIST CAVP, CCTL has performed all assurance/evaluation activities and documented in the ETR and AAR accordingly.	N/A. This test has been successfully tested in FTP_DIT_EXT.1 that uses safe-prime groups.
FCS_COP.1/SKC	AES used in [**CBC, CTR**] (as defined in NIST SP 800-38A) mode and	Microsoft Windows 10 Anniversary Update, Windows Server 2016,	AES-CBC AES-CTR	AES #4064

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
	cryptographic key sizes [**128 bits, 256 bits**]	Windows Storage Server 2016; Microsoft Surface Book, Surface Pro 4, Surface Pro 3 and Surface 3 w/ Windows 10 Anniversary Update; Microsoft Lumia 950 and Lumia 650 w/ Windows 10 Mobile Anniversary Update SymCrypt Cryptographic Implementations		
FCS_COP.1/ Sig	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.	Microsoft Windows 10 Anniversary Update, Windows Server 2016, Windows Storage Server 2016; Microsoft Surface Book, Surface Pro 4, Surface Pro 3 and Surface 3 w/ Windows 10 Anniversary Update; Microsoft Lumia 950 and Lumia 650 w/ Windows 10 Mobile Anniversary Update MsBignum Cryptographic Implementations	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	RSA #2193
		Microsoft Windows 10 Anniversary Update, Windows Server 2016, Windows Storage Server 2016; Microsoft Surface Book, Surface Pro 4, Surface Pro 3 and Surface 3 w/ Windows 10 Anniversary Update; Microsoft Lumia 950 and	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	RSA #2192

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
		Lumia 650 w/ Windows 10 Mobile Anniversary Update Cryptography Next Generation (CNG) Implementations		
	For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384, and P521]	Microsoft Windows 10 Anniversary Update, Windows Server 2016, Windows Storage Server 2016; Microsoft Surface Book, Surface Pro 4, Surface Pro 3 and Surface 3 w/ Windows 10 Anniversary Update; Microsoft Lumia 950 and Lumia 650 w/ Windows 10 Mobile Anniversary Update MsBignum Cryptographic Implementations	ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)	ECDSA #911
FCS_COP.1/ Hash	[**SHA-1, SHA-256, SHA-384, SHA-512**] and message digest sizes [**160, 256, 384, 512**] bits	Microsoft Windows 10 Anniversary Update, Windows Server 2016, Windows Storage Server 2016; Microsoft Surface Book, Surface Pro 4, Surface Pro 3 and Surface 3 w/ Windows 10 Anniversary Update; Microsoft Lumia 950 and Lumia 650 w/ Windows 10 Mobile Anniversary Update SymCrypt Cryptographic Implementations	SHA-1 SHA2-256 SHA2-384 SHA2-512	SHS #3347

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
FCS_COP.1/ KeyedHash	[**HMAC-SHA- 256, HMAC-SHA-384, HMAC-SHA-512**] and cryptographic key sizes [**key size (in bits) used in HMAC**] and message digest sizes [**256, 384, 512**] bits	Microsoft Windows 10 Anniversary Update, Windows Server 2016, Windows Storage Server 2016; Microsoft Surface Book, Surface Pro 4, Surface Pro 3 and Surface 3 w/ Windows 10 Anniversary Update; Microsoft Lumia 950 and Lumia 650 w/ Windows 10 Mobile Anniversary Update SymCrypt Cryptographic Implementations	HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512	HMAC #2651
FCS_RBG_EXT.1	invoke platform-provided DRBG functionality	Microsoft Windows 10 Anniversary Update, Windows Server 2016, Windows Storage Server 2016; Microsoft Surface Book, Surface Pro 4, Surface Pro 3 and Surface 3 w/ Windows 10 Anniversary Update; Microsoft Lumia 950 and Lumia 650 w/ Windows 10 Mobile Anniversary Update SymCrypt Cryptographic Implementations	Counter DRBG	DRBG #1217