Trellix Endpoint Security (HX) Agent v35.31.31 Security Target

Document Version: 2.3

intertek acumen security

2400 Research Blvd

Suite 395

Rockville, MD 20850

Revision History:

Version	Date	Changes
Version 0.1	September 1, 2021	Initial Draft
Version 0.2	October 1, 2021	Updated based on internal reviews
Version 0.3	October 14, 2021	Updated based on QA comments
Version 0.4	June 14, 2022	Updated PP v1.3 -> v1.4, addressing review comments
Version 0.5	June 24, 2022	Updated based on a call with the developer
Version 0.6	June 29, 2022	Updated based on a review and call with the developer
Version 0.7	July 12, 2022	Removed Linux platforms
Version 0.8	July 26, 2022	Updated TOE and developer identifiers
Version 0.9	December 22, 2022	Incorporated TDs
Version 1.0	May 17, 2023	Incorporated new TDs
Version 1.1	June 27, 2023	Removed HTTPS SFR
Version 1.2	June 27, 2023	Addressed QA comments
Version 1.3	July 25, 2023	Added back HTTPS SFR
Version 1.4	August 21, 2023	Addressed evaluator comments, added support for TLS Client Support for Renegotiation.
Version 1.5	September 6, 2023	Applied TD0747 and archived TD0624
Version 1.6	October 2, 2023	Applied TD0779 and archived TD0588
Version 1.7	October 04, 2023	Addressed equivalency report comments
Version 1.8	October 12, 2023	Addressing ECR comments of the check-in package
Version 1.9	October 17, 2023	Addressing ECR comments of the check-in package
Version 2.0	January 16, 2024	Updated sections 1.4.2 and 2.3.1
Version 2.1	April 3, 2024	Addressing reviewer comments
Version 2.2	May 10, 2024	Addressing validators ECR comments
Version 2.3	May 21, 2024	Addressing validators second round of ECR comments

Contents

1 Introduction

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document describes the intended operational environment of the TOE, as well as the security functional and assurance requirements that are met by the TOE.

1.1 Security Target and TOE Reference

This section provides the information needed to identify and control the TOE and the ST. The ST and the TOE are identified as per information given in Table 1.

Table 1 – TOE/ST Identification

TOE/ST Identifier	Identifier Value
ST Title	Trellix Endpoint Security (HX) Agent v35.31.31 Security Target
ST Version	2.3
ST Date	May 21, 2024
ST Author	Acumen Security, LLC.
TOE Identifier	Trellix Endpoint Security (HX) Agent
TOE Version	v35.31.31
TOE Developer	Trellix US LLC,
Key Words	FireEye, Trellix, X-Agent, software, TLS

1.2 Product Overview

The TOE: the Endpoint Security (HX) xAgent protects enterprise networks by monitoring each endpoint device or host, collecting real-time data of events occurring on the endpoint, and identifying threat activity and evidence on the host where it is installed. Threat activity and evidence include:

- Unauthorized use of valid accounts
- Trace evidence and partial files
- Command and control activity
- Known and unknown malware
- Suspicious network traffic
- Valid programs used for malicious purposes.
- Unauthorized file access
- Exploits and other online attacks (found using xAgent Exploit Guard functionality)
- Commodity malware (found using xAgent malware protection functionality)

When the TOE finds evidence of potential compromises, it reports this information to the Endpoint Security server. It also retrieves information and tasks (jobs) from the Endpoint Security (HX) server and performs them. Tasks include upgrading indicators of compromise, requests for forensic information (file, triage, and data requests), and requests to contain the host endpoint.

Endpoint Security (HX) xAgent can be provision to an on-premises, virtual, or cloud Endpoint Security server.

1.3 TOE Overview

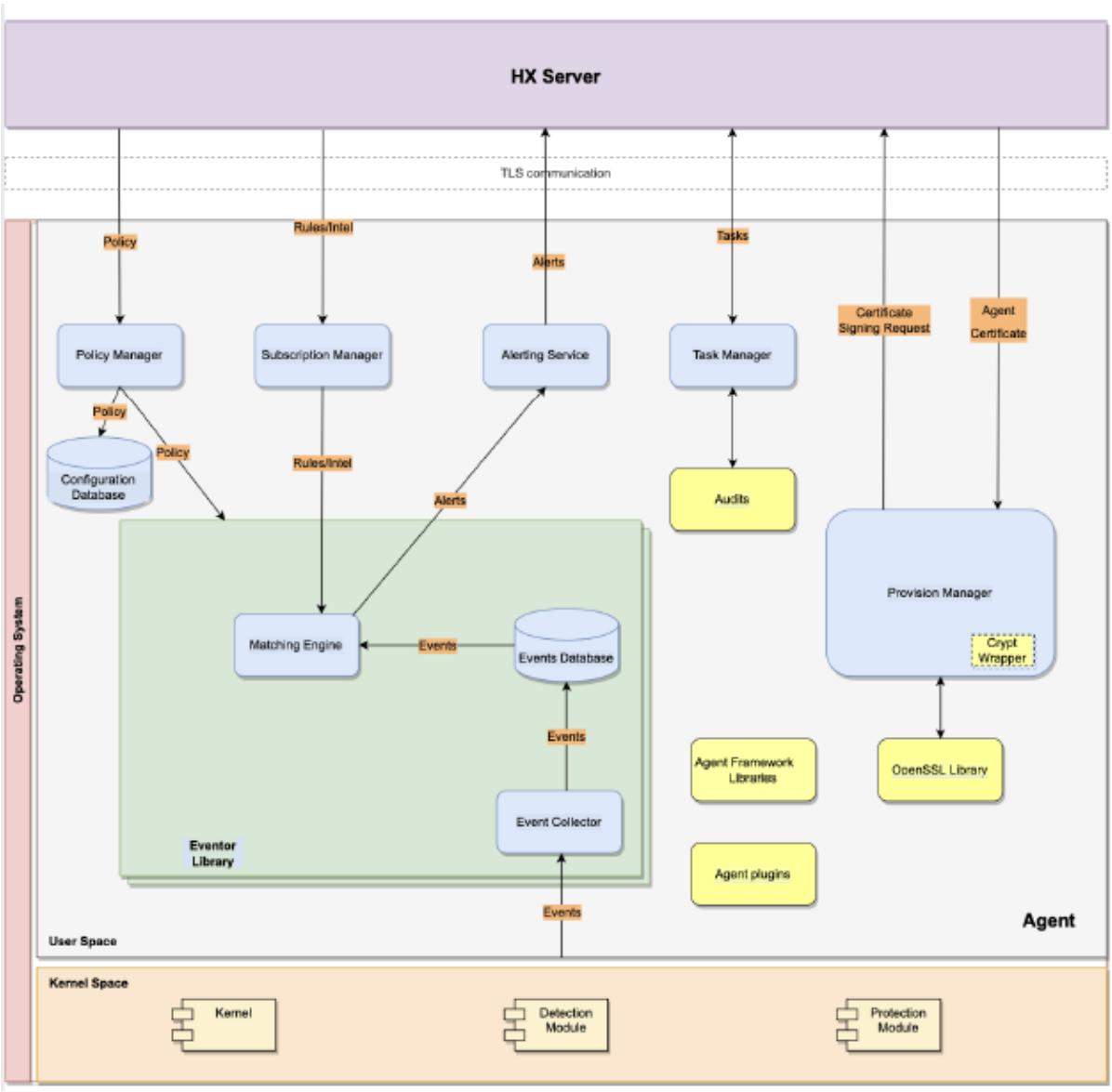
The TOE is the Trellix Endpoint Security (HX) Agent v35.31.31, a software application residing on a host platform and interacting exclusively with a Trellix Endpoint Security (HX) Series appliances. The TOE is an enterprise-managed agent that runs in the background of the host platform of an endpoint to provide protection against common malware as well as advanced attack. Based on a defense in depth model, the TOE uses a modular architecture with default engines and downloadable modules to protect, detect and respond to security events. There are no users interacting with the TOE or being informed of any communication between the TOE and the HX Series appliance.

1.4 TOE Description

This section provides an overview of the TOE, including physical boundary, the security functions implemented by the TOE, and any relevant TOE documentation and references.

A representative deployment of the TOE is illustrated in Figure 1. TOE is a software agent executing on a host platform and interacting with the Trellix Endpoint Security (HX) Server. The TOE operates predominantly in the user space with the exception of some event sources requiring interaction with the kernel space of the host platform. The communication between the TOE and the HX Series Appliance (i.e., HX Server) is protected with TLS.

Figure 1 –TOE Structure and Deployment



1.4.1 Physical Boundary

The TOE is packaged with 32-bit libraries and some 64-bit versions of libraries, making the TOE platform-agnostic application software that can be executed on all the claimed TOE's platforms. The host platform is not part of the TOE. When deployed, the TOE is pushed to the host platform from a Trellix Endpoint Security (HX) Series appliance. It installs natively as a kernel and user space application.

The TOE runs on the following Microsoft Windows Operating Systems which is running on VMware hypervisor 7.0 on an Intel Xeon E5-4620 V4 processor (Broadwell microarchitecture), which are the only allowed host platforms:

- Windows 10 Version 21H2 32-bits running on ESXi Hypervisor v7.0 on an Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows 10 Version 1803 32-bits running on ESXi Hypervisor v7.0 on an Intel Xeon E5-4620 V4 processor (Broadwell).

- Windows 10 Version 1903 32-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows 10 Version 1909 LTSC 32-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows 10 Version 2004 32-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows 10 Version 21H2 64-bits running on ESXi Hypervisor v7.0 on an Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows 10 Version 1803 64-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows 10 Version 1903 64-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows 10 Version 1909 LTSC 64-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows 10 Version 2004 64-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows 11 Version 21H2 64-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows Server 2016 running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows Server 2019 running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows Server 2012 R2 running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows Server 2008 R2 (SP1) running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).
- Windows Server 2022 running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).

1.4.2 Security Functions Provided by the TOE

The TOE implements all security functions and mechanisms required for conformance with [PP_APP_v1.4] and [PKG_TLS_V1.1][1].

1.4.2.1 Cryptographic Support

The TOE implements cryptographic support for the following:

- TLS connectivity between itself and a Trellix Endpoint Security (HX) Series Appliance, including generation of 2048-bit RSA keys for a certificate signing request and implementation of all required cryptographic algorithms, and
- Digital certificate validation.

The cryptographic algorithms the TOE implements and the CAVP certificate numbers are given in Table 2. Each algorithm is implemented using the OpenSSL Cryptographic Library version 3.0.8 which is part of the TOE.

[1] See Sect. 2.2.

Table 2 TOE Cryptographic Algorithms and CAVP Certificate References

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
FCS_CKM.1/AK	RSA schemes using cryptographic key sizes of 2048-bit that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	Trellix OpenSSL FIPS Provider v3.0.8	RSA KeyGen (FIPS186-4)	A5228
FCS_CKM.2	RSA key establishment schemes that meet the following: NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography"	Trellix OpenSSL FIPS Provider v3.0.8	Vendor Affirmed	Vendor Affirmed
FCS_COP.1/ SKC	AES-CBC mode as defined in NIST SP 800-38A and cryptographic key sizes 128 bits and 256 bits	Trellix OpenSSL FIPS Provider v3.0.8	AES-CBC	A5228
FCS_COP.1/ Hash	SHA-1 and SHA-256 and message digest sizes 160 and 256 bits	Trellix OpenSSL FIPS Provider v3.0.8	SHA-1 SHA2-256	A5228
FCS_COP.1/ Sig	RSA scheme using cryptographic key sizes of 2048-bit that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5	Trellix OpenSSL FIPS Provider v3.0.8	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	A5228
FCS_COP.1/ KeyedHash	HMAC-SHA-1 and HMAC-SHA-256 with key sizes 256 and 160 bits used in HMAC and message digest sizes 256 and 160 bits that meet the following: FIPS Pub 198-1,'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'	Trellix OpenSSL FIPS Provider v3.0.8	HMAC-SHA-1 HMAC-SHA2- 256	A5228
FCS_RBG_EXT.2.1	An NIST Special Publication 800-90A using CTR_DRBG(AES) with a minimum of 256-bits	Trellix OpenSSL FIPS Provider v3.0.8	Counter DRBG	A5228

1.4.2.2 Identification and Authentication

The TOE uses X.509v3 certificates as defined by RFC 5280 to authenticate the TLS connection to the Trellix Endpoint Security (HX) Series appliance. The TOE validates the X.509 certificates using the certificate path validation algorithm defined in RFC 5280.

1.4.2.3 User Data Protection

The TOE is distributed as an installer package in Microsoft Installer (MSI) format. As well as the initial installation package, all updates to the TOE are also distributed as MSI packages. Each TOE installation and update package is digitally signed by Trellix in the production environment of the TOE. There are several methods to acquire the TOE's installation images. These include downloading them from the HX server, manually obtaining them from the vendor's cloud servers, or accessing them from the vendor's offline portal. Subsequent updates for the TOE can either be distributed from the HX server or downloaded and installed manually on the host machine.

1.4.2.4 Privacy

The TOE does not transmit Personally Identifiable Information (PII) over the network. This protects the privacy of the users of the host platform.

1.4.2.5 Protection of the TSF

The TOE implements several security mechanisms to protect itself when installed on the host platform. Protection of the installation and update packages when stored on the Trellix Endpoint Security (HX) Series appliance or on the TOE is using digital signatures as described in Sect. 1.4.2.3.

The TOE never allocates memory with both write and execute permissions. Furthermore, the TOE operates in an environment in which the following security mechanisms 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.

Protection of the TOE and parts of it when stored within the production environment is not in the scope of the evaluation. Nevertheless, during compilation, the TOE is built with several flags enabled to check for engineering flaws. The flags and the protection mechanisms include the following:

- The TOE is built with the /GS flag enabled. This reduces the possibilities of stack-based buffer overflows in the product.
- The compiler enables Address Space Layout Randomization (ASLR) by default.
- The TOE is not built with the /DYNAMICBASE:NO which would disable ASLR.

1.4.2.6 Trusted Path/Channels

The TOE receives scanning policies from the associated Trellix Endpoint Security (HX) Series appliance over a network connection. The TOE uses the scanning policies for scanning the host platform and returns the results of the scanning to the appliance. The connection between the TOE and the Trellix Endpoint Security (HX) Series appliance is always secured with TLS. The TLS is implemented in full conformance with [PKG_TLS_V1.1].

1.4.3 TOE Documentation

The following documents are essential to understanding and controlling the TOE in the evaluated configuration:

- Trellix Endpoint Security (HX) Agent v35.31.31 Security Target, version 2.3
- Trellix Endpoint Security (HX) Agent v35.31.31 Common Criteria Guidance Supplement, version 1.4
- Endpoint Security xAgent Deployment Guide Release 35.31.0

1.5 Environment

The following environmental components are required to operate the TOE in the evaluated configuration:

Table 3 – Required Environmental Components

Component	Description
Trellix Endpoint Security (HX) Server	Trellix Endpoint Security (HX) Server is the server from which the TOE and updates thereof are installed on host platforms. For installation on a host platform, the TOE and any updates thereof need to be uploaded from the production environment to the Trellix Endpoint Security (HX) server. The HX server UI is used to deploy the TOE and configure many of its configurations and settings. Also, the HX server acts as a PKI server, which creates and signs all the used X509 certificates deployed to the TOE. The TOE collects system events (file, process, registry, network etc.) and processes them as per business logic expressed as scanning rules. It then communicates the results of the scanning to the Trellix Endpoint Security (HX) Server. The Trellix Endpoint Security (HX) Server implements HTTPS TLS for secure communication between itself and the TOE and uses that for all communication.
Host Platform	The Host Platform may be any computer with an allowed Microsoft Windows operating system. The host platform must have at least 1GB of system memory. The Host Platform must also implement the necessary network connectivity for the TOE to communicate with the Trellix Endpoint Security (HX) Server. While the TOE implements TLS to protect the content of the communication, the Host Platform must implement the protocol stacks and the physical ports for the connectivity.
CRL Server	The TOE must be associated to a Certificate Revocation List (CRL) Server. The CRL Server contains the revocation list which is communicated to the TOE and used in the validation of the X.509 certificates. The CRL Server is part of the management server associated to the Trellix Endpoint Security (HX) Server.

1.6 Product Functionality not Included in the Scope of the Evaluation

The following product functionality is not included in the CC evaluation:

- SHA-1 is used only in the provisioning of the TOE, not in the digital signature and session authentication functions implemented by the TOE.
- xAgent to HX server communication using fast-pooling check on TCP port 80.
- Real-Time Indicator Detection.
- Trellix Exploit Guard Protection.
- Malware Protection.
- The scanning functions, or the specifics of the scanning policies and how they are managed.

2 Conformance Claims

This section identifies the TOE conformance claims, conformance rationale, and relevant Technical Decisions (TDs).

2.1 CC Conformance Claims

The ST and the TOE are Common Criteria conformant to the following:

- 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, and
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017.

This ST is Common Criteria Part 2 Extended conformant and Common Criteria Part 3 extended conformant.

This ST is package-conformant to the following package: [PKG_TLS_V1.1] Functional Package for Transport Layer Security, Version 1.1, March 1, 2019.

2.2 Protection Profile Conformance

This ST also claims exact conformation to the following protection profile:

[PP_APP_v1.4] Protection Profile for Application Software, Version 1.4, 2021-10-07

2.3 Conformance Rationale

This ST claims exact conformance to [PP_APP_v1.4] and [PKG_TLS_V1.1]. The security problem definition and the statement of security objectives are taken from them unmodified.

The statement of security requirements is taken from [PP_APP_v1.4] and [PKG_TLS_V1.1]. Only operations permitted therein are implemented. Selection-based and optional requirements (if any) are in conformance with [PP_APP_v1.4] and [PKG_TLS_V1.1].

2.3.1 Technical Decisions

All NIAP Technical Decisions (TDs) issued to date and applicable to [PP_APP_v1.4] and [PKG_TLS_V1.1] have been considered. Table 4 identifies all applicable TDs and states their applicability to the ST. Any exclusion is justified in the exclusion rationale.

Table 4 – Relevant Technical Decisions applicable to the ST

Technical Decision	Applicable	Exclusion Rationale (where applicable)
PP_APP_v1.4: Active Related Technical Decisions		
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	
TD0815: Addition of Conditional TSS Activity for FPT_AEX_EXT.1.5	Yes	
TD0798: Static Memory Mapping Exceptions	Yes	
TD0780: FIA_X509_EXT.1 Test 4 Clarification	Yes	
TD0756 – Update for platform-provided full disk encryption	Yes	
TD0747: Configuration Storage Option for Android	No	TOE is based on Windows platform
TD0743: FTP_DIT_EXT.1.1 Selection exclusivity	Yes	
TD0736: Number of elements for iterations of FCS_HTTPS_EXT.1	No	Toe does not claim FCS_HTTPS_EXT.1/Server
TD0719: ECD for PP APP V1.3 and 1.4	Yes	
TD0717: Format changes for PP_APP_V1.4	Yes	
TD0664: Testing activity for FPT_TUD_EXT.2.2	Yes	
TD0650: Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4	No	ST does not claim PP-Module for VPN Clients, Version 2.4
TD0628: Addition of Container Image to Package Format	Yes	
PKG_TLS_v1.1: Active Related Technical Decisions		
TD0779: Updated Session Resumption Support in TLS package V1.1	No	ST does not claim TLS server
TD0770: TLSS.2 connection with no client cert	No	ST does not claim TLS server
TD0739: PKG_TLS_V1.1 has 2 different publication dates	No	ST does not claim TLS server
TD0726: Corrections to (D)TLSS SFRs in TLS 1.1 FP	No	ST does not claim TLS server
TD0513: CA Certificate loading	Yes	
TD0499: Testing with pinned certificates	Yes	
TD0469: Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	No	The TOE does not implement TLS Server
TD0442: Updated TLS Ciphersuites for TLS Package	Yes	

3 Security Problem Definition

The security problem definition is taken directly from the claimed PP and any relevant EPs/Modules/Packages identified in Section 2.2 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 threats included in Table 5 are drawn directly from the PP and any EPs/Modules/Packages identified in Section 2.2.

Table 5 – Threats

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.

3.2 Assumptions

The assumptions included in Table 6 are drawn directly from PP and any relevant EPs/Modules/Packages.

Table 6 – Assumptions

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 with the applied enterprise security policy.

3.3 Organizational Security Policies

There are no OSPs defined for the TOE.

4 Security Objectives

The security objectives have been taken directly from the claimed PP and any relevant EPs/Modules/Packages and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the TOE

The security objectives in the following table apply to the TOE.

Table 7 – Security Objectives

ID	Security Objectives
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. 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.
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.
O.MANAGMENT	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.
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.
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.

4.2 Security Objectives for the Operational Environment

Security objectives for the operational environment assist the TOE in correctly providing its security functionality. These objectives, which are found in the table below, track with the assumptions about the TOE operational environment.

Table 8 – Security Objectives for the Operational Environment

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

5 Extended Components Definition

5.1 Extended Security Functional Components

All extended components are sourced directly from [PP].

5.2 Extended Security Functional Requirements Rationale

All extended security functional components are sourced directly from [PP]. Exact conformance required by the PP also mandates inclusion of all applicable extended components defined in the PP.

6 Security Requirements

This section identifies and states the Security Functional Requirements (SFRs) fulfilled by the TOE. The SFRs included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, September 2017, applicable Protection Profiles and Functional Packages, and all applicable international interpretations. Where required, the statements of the SFRs are modified in accordance with the applicable NIAP Technical Decisions.

The Security Functional Components fulfilled by the TOE are summarized in Table 9.

Table 9 – Security Functional Components

Functional Component	Description
FCS_CKM_EXT.1	Cryptographic Key Generation Services
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/Sig	Cryptographic Operation – Signing
FCS_COP.1/KeyedHash	Cryptographic Operation – Keyed-Hash Message Authentication
FCS_RBG_EXT.1	Random Bit Generation Services
FCS_RBG_EXT.2	Random Bit Generation from Application
FCS_STO_EXT.1	Storage of Credentials
FCS_HTTPS_EXT.1/Client	HTTPS Protocol
FCS_HTTPS_EXT.2	HTTPS Protocol and Mutual Authentication
FCS_TLS_EXT.1	TLS Protocol
FCS_TLSC_EXT.1	TLS Client Protocol
FCS_TLSC_EXT.2	TLS Client Support for Mutual Authentication
FCS_TLSC_EXT.4	TLS Client Support for Renegotiation
FDP_DEC_EXT.1	Access to Platform Resources
FDP_NET_EXT.1	Network Communications
FDP_DAR_EXT.1	Encryption of Sensitive Application Data
FIA_X509_EXT.1	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FMT_CFG_EXT.1	Secure by Default Configuration
FMT_MEC_EXT.1	Supported Configuration Mechanism
FMT_SMF.1	Specification of Management Functions
FPR_ANO_EXT.1	User Consent for Transmission of Personally Identifiable Information
FPT_AEX_EXT.1	Anti-Exploitation Capabilities
FPT_API_EXT.1	Use of Supported Services and APIs
FPT_IDV_EXT.1	Software Identification and Versions
FPT_LIB_EXT.1	Use of Third-Party Libraries

Functional Component	Description
FPT_TUD_EXT.1	Integrity for Installation and Update
FPT_TUD_EXT.2	Integrity for Installation and Update
FTP_DIT_EXT.1	Protection of Data in Transit

6.1 Conventions

The CC allows the following types of operations to be performed on the functional requirements: assignments, selections, refinements, and iterations. The following font conventions are used within this document to identify operations defined by CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending the iteration identifier followed by an iteration description, e.g., FCS_COP.1/Sig.
- Where operations were completed in the PP and relevant EPs/Modules/Packages, the formatting used in the PP has been retained.
- Extended SFRs and SARs are identified by the addition of "EXT" after the requirement name.

6.2 Security Functional Requirements

This section includes the security functional requirements for this ST.

6.2.1 Cryptographic Support (FCS)

6.2.1.1 FCS_CKM_EXT.1 Cryptographic Key Generation Services

FCS_CKM_EXT.1.1

The **application** shall [

- *implement asymmetric key generation*

].

Application Note: Modified according to TD0717.

6.2.1.2 FCS_CKM.1/AK Cryptographic Asymmetric Key Generation

FCS_CKM.1.1/AK

The application shall [

- *implement functionality*

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

[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"***

].

Application Note: Modified according to TD0717.

6.2.1.3 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1

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

- ***[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"]]***

6.2.1.4 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,*

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

Application Note: Modified according to TD0717.

6.2.1.5 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*

] and message digest sizes [

- *160,*
- *256*

] bits that meet the following: [FIPS PUB 180-4].

Application Note: Modified according to TD0717.

6.2.1.6 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]*

].

Application Note: Modified according to TD0717.

6.2.1.7 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*

]and [

- *HMAC-SHA-1*

] with key sizes [*256 and 160 bits used in HMAC*] and message digest sizes [256] and [*160*] bits that meet the following: [FIPS Pub 198-1,'*The Keyed-Hash Message Authentication Code'* and FIPS Pub 180-4 '*Secure Hash Standard'].*

Application Note: Modified according to TD0717.

6.2.1.8 FCS_HTTPS_EXT.1/Client HTTPS Protocol

FCS_HTTPS_EXT.1.1/Client

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

FCS_HTTPS_EXT.1.2/Client

The application shall implement HTTPS using TLS as defined in the TLS package.

FCS_HTTPS_EXT.1.3/Client

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

6.2.1.9 FCS_HTTPS_EXT.2 HTTPS Protocol with Mutual Authentication

FCS_HTTPS_EXT.2.1

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

6.2.1.10 FCS_RBG_EXT.1 Random Bit Generation Services

FCS_RBG_EXT.1.1

The application shall [

- *implement DRBG functionality*

] for its cryptographic operations.

6.2.1.11 FCS_RBG_EXT.2 Random Bit Generation from Application

FCS_RBG_EXT.2.1

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

FCS_RBG_EXT.2.2

The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [

- *no other noise source*

] with a minimum of [

- *256 bits*

] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

6.2.1.12 FCS_STO_EXT.1 Storage of Credentials

FCS_STO_EXT.1.1

The application shall [

- *implement functionality to securely store [private key, digital certificates]*

] according to [FCS_COP.1/SKC]] to non-volatile memory.

6.2.1.13 FCS_TLS_EXT.1 TLS Protocol

FCS_TLS_EXT.1.1

The product shall implement [

- TLS as a client

].

6.2.1.14 FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1

The product shall implement TLS 1.2 (RFC 5246) and [no earlier TLS versions*]* as a client that supports the cipher suites [

- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246*

] and also supports functionality for [

- *mutual authentication*
- *session renegotiation*

].

Application Note: Modified according to TD0442.

FCS_TLSC_EXT.1.2

The product shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3

The product shall not establish a trusted channel if the server certificate is invalid [

- *with no exceptions*

].

6.2.1.15 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

FCS_TLSC_EXT.2.1

The product shall support mutual authentication using X.509v3 certificates.

6.2.1.16 FCS_TLSC_EXT.4 TLS Client Support for Renegotiation

FCS_TLSC_EXT.4.1

The product shall support secure renegotiation through use of the “renegotiation_info” TLS extension in accordance with RFC 5746.

6.2.2 User Data Protection (FDP)

6.2.2.1 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.

6.2.2.2 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,*
- *[RAM, filesystem]*

].

6.2.2.3 FDP_NET_EXT.1 Network Communications

FDP_NET_EXT.1.1

The application shall restrict network communication to [

- *[polling and downloading new scanning policies to be used to identify potential intrusions on the host OS from the associated Trellix Endpoint Security (HX) appliance, sending information to the associated Trellix Endpoint Security (HX) appliance as defined in the downloaded scanning policies]*

].

6.2.3 Security Requirements (FIA)

6.2.3.1 FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1

The application shall [*implement functionality*] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The application shall validate that any CA certificate includes caSigning purpose in the key usage field.
- The application shall validate the revocation status of the certificate using [*CRL as specified in RFC 5280 Section 6.3*].
- 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 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 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.

6.2.3.2 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 [HTTPS, *TLS*].

FIA_X509_EXT.2.2

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

Application Note: Please refer to the TSS section for this SFR for further clarification on this selection.

6.2.4 Security Management (FMT)

6.2.4.1 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].

6.2.4.2 FMT_CFG_EXT.1 Secure by Default Configuration

FMT_CFG_EXT.1.1

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

FMT_CFG_EXT.1.2

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

6.2.4.3 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1

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

- *Allowing Windows exploit guard protection.*

].

6.2.5 Privacy (FPR)

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

FPR_ANO_EXT.1.1

The application shall [

- *not transmit PII over a network*

].

6.2.6 Protection of the TSF (FPT)

6.2.6.1 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 built with stack-based buffer overflow protection enabled.

6.2.6.2 FPT_API_EXT.1 Use of Supported Services and APIs

FPT_API_EXT.1.1

The application shall use only documented platform APIs.

6.2.6.3 FPT_IDV_EXT.1 Software Identification and Versions

FPT_IDV_EXT.1.1

The application shall be versioned with *[SWID tags that comply with minimum requirements from ISO/IEC 19770-2:2015]*.

6.2.6.4 FPT_LIB_EXT.1 Use of Third-Party Libraries

FPT_LIB_EXT.1.1

The application shall be packaged with only [

- *Libuv.dll,*
- *fips.dll*
- *legacy.dll*
- *libcrypto-3-x64.dll*
- *libssl-3-x64.dll*
- *libssl-3.dll*
- *libcrypto-3.dll*
- *visual studio runtime libraries (vcruntime140.dll, vccorlib140.dll, msvcp140.dll, concrt140.dll, ucrtbase.dll, api-ms-win-core-console-l1-1-0.dll, api-ms-win-core-console-l1-2-0.dll, api-ms-win-core-datetime-l1-1-0.dll, api-ms-win-core-debug-l1-1-0.dll, api-ms-win-core-errorhandling-l1-1-0.dll, api-ms-win-core-fibers-l1-1-0.dll, api-ms-win-core-file-l1-1-0.dll, api-ms-win-core-file-l1-2-0.dll, api-ms-win-core-file-l2-1-0.dll, api-ms-win-core-handle-l1-1-0.dll, api-ms-win-core-heap-l1-1-0.dll, api-ms-win-core-interlocked-l1-1-0.dll, api-ms-win-core-libraryloader-l1-1-0.dll, api-ms-win-core-localization-l1-2-0.dll, api-ms-win-core-memory-l1-1-0.dll, api-ms-win-core-namedpipe-l1-1-0.dll, api-ms-win-core-processenvironment-l1-1-0.dll, api-ms-win-core-processthreads-l1-1-0.dll, api-ms-win-core-processthreads-l1-1-1.dll, api-ms-win-core-profile-l1-1-0.dll, api-ms-win-core-rtlsupport-l1-1-0.dll, api-ms-win-core-string-l1-1-0.dll, api-ms-win-core-synch-l1-1-0.dll, api-ms-win-core-synch-l1-2-0.dll, api-ms-win-core-sysinfo-l1-1-0.dll, api-ms-win-core-timezone-l1-1-0.dll, api-ms-win-core-util-l1-1-0.dll, api-ms-win-crt-conio-l1-1-0.dll, api-ms-win-crt-convert-l1-1-0.dll, api-ms-win-crt-environment-l1-1-0.dll, api-ms-win-crt-filesystem-l1-1-0.dll, api-ms-win-crt-heap-l1-1-0.dll, api-ms-win-crt-locale-l1-1-0.dll, api-ms-win-crt-math-l1-1-0.dll, api-ms-win-crt-*

multibyte-l1-1-0.dll, api-ms-win-crt-private-l1-1-0.dll, api-ms-win-crt-process-l1-1-0.dll, api-ms-win-crt-runtime-l1-1-0.dll, api-ms-win-crt-stdio-l1-1-0.dll, api-ms-win-crt-string-l1-1-0.dll, api-ms-win-crt-time-l1-1-0.dll, api-ms-win-crt-utility-l1-1-0.dll, msvcp140_1.dll, msvcp140_2.dll, msvcp140_atomic_wait.dll, msvcp140_codecvt_ids.dll, vcruntime140_1.dll)),

- *zlib1.dll*

].

6.2.6.5 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*]

6.2.6.6 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.*]

Application Note: Modified according to TD0628.

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.

6.2.7 Trusted Path/Channel (FTP)

6.2.7.1 FTP_DIT_EXT.1 Protection of Data in Transit

FTP_DIT_EXT.1.1

The application shall [

- *encrypt all transmitted [data] with [*
  - *HTTPS as a client in accordance with FCS_HTTPS_EXT.1/Client for [exchanging data],*
  - *TLS as a client as defined in the Functional Package for TLS for [mutual authentication]]*

] between itself and another trusted IT product.

Application Note: Modified according to TD0743.

6.3 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the PP and the relevant ones derived from Common Criteria Version 3.1, Revision 5. The assurance components applicable to the TOE are summarized in Table 10.

Table 10 – Security Assurance Requirements

Assurance Class	Assurance Components	Component Description
Security Target	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	Security problem definition
	ASE_TSS.1	TOE summary specification
Development	ADV_FSP.1	Basic functionality specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative user guidance
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	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 survey

6.4 Assurance Measures

The TOE satisfies the assurance requirements summarized in Sect. 6.3. This section states the Assurance Measures applied by Trellix US LLC, to satisfy the assurance requirements. The following table lists the details.

Table 11 TOE Security Assurance Measures

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

SAR Component	How the SAR will be met
	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 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	Users of the TOE should report any security related issues via the Trellix webpage (https://www.trellix.com/support.html), which provides a secure channel. Software updates/fixes are also provided via the Trellix webpage. Public availability of an update for a publicly disclosed vulnerability is typically 90 days or less and a maximum of 180 days.
ATE_IND.1	Trellix will provide the TOE for testing.
AVA_VAN.1	Trellix will provide the TOE for testing. Trellix will provide a document identifying the list of software and hardware components.

7 TOE Summary Specification

Table 12 describes how the Security Functional Requirements and the assurance component ALC_TSU_EXT.1 are met by the TOE.

Cryptographic algorithms implemented by the TOE and the respective CAVP Certificate references are given in Table 2.

Table 12 – TOE Summary Specification

Requirement	TSS Description
ALC_TSU_EXT.1	Users of the TOE should report any security related issues via the Trellix webpage (https://www.trellix.com/en-us/support/fe-support.html [trellix.com]). The webpage implements a secure channel for the reporting. Software updates/fixes are also provided by the developer via the Trellix webpage. Public availability of an update for a publicly disclosed vulnerability is typically 90 days or less and a maximum of 180 days.
FCS_CKM_EXT.1 FCS_CKM.1/AK	The TOE shall generate a 2048-bit RSA public-private key pair and construct a Certificate Signing Request (CSR) with that key pair. The CSR is sent to the Trellix Endpoint Security (HX) Series Appliance which constructs an X.509 certificate from the CSR and returns it to the TOE.
FCS_CKM.2	For RSA Key Establishment, the TOE implements sections 7.1 and 7.2.1 of SP 800-56B. The TOE does not perform any operation marked as “Shall Not” or “Should not” in SP 800-56B. Additionally, the TOE does not omit any operation marked as “Shall.” The TOE implements a TLS client (i.e., sender) and, therefore, does not perform RSA decryption. NO CAVP certificate exists, and Vendor affirmation is claimed.
FCS_COP.1/SKC	The TOE implements symmetric encryption and decryption using AES in CBC mode (128 bits and 256 bits) as described in NIST SP 800-38A.
FCS_COP.1/Hash	The TOE implements cryptographic hashing using SHA-1 and SHA-256 with message digest sizes 160 and 256 bits respectively. Implementation is in accordance with FIPS Pub 180-4 “Secure Hash Standard.” SHA-1 is only used in the provisioning of the TOE, not in the digital signature functions. SHA-256 is used in TLS session negotiation and with HMACs used to verify the integrity of TLS traffic. SHA-256 is also used in conjunction with RSA as part of the Trellix Endpoint Security (HX) Series Appliance X.509 certificate verification.
FCS_COP.1/Sig	The TOE implements cryptographic signature computation using RSA Digital Signature Algorithm with key size of 2048 as specified in FIPS PUB 186-4, “Digital Signature Standard”.
FCS_COP.1/KeyedHash	The TOE implements keyed-hashing message authentication function using HMAC-SHA-256 and HMAC-SHA-1 with key sizes and message digests size of 256 and 160 bits. The implementation is as specified in FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code,” and FIPS 180-4, “Secure Hash Standard.”
FCS_HTTPS_EXT.1/Client FCS_HTTPS_EXT.2	The TOE implements the HTTPS protocol according to RFC 2818 by implementing all SHALL, MUST, and SHOULD statements and by not implementing any SHALL NOT,

Requirement	TSS Description
	MUST NOT, or SHOULD NOT statements. HTTPS is implemented using TLS 1.2 (RFC 5246). The TOE's interface does not accept a connection when a peer's certificate is invalid.
FCS_RBG_EXT.1 FCS_RBG_EXT.2	The TOE implements random bit generation services using an SP 800-90A CTR_DRBG using AES-256. The DRBG is seeded with at least 256 bits of entropy from the DRBG provided by the Windows platform through the CryptGenRandom API.
FCS_STO_EXT.1	The TOE stores digital certificates and private keys in the TOE's JSON structure stored in non-volatile memory. The database is encrypted by the TOE with AES and is not accessible to any external entity. The certificates are used to validate the HX server certificate. The RSA key pairs are used for encryption, decryption, and digital signatures during TLS communication with only the HX server.
FCS_TLS_EXT.1 FCS_TLSC_EXT.1 FCS_TLSC_EXT.2 FCS_TLSC_EXT.4	In support of secure communication with external entities, the TOE implements the TLS protocol using mutual authentication mechanism. TLS is used to facilitate communication with the Trellix Endpoint Security (HX) Series Appliances. The TOE only communicates with the Trellix Endpoint Security (HX) appliance using: – TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246. X.509 certificates used for this connection are validated using the certificate path validation algorithm defined in RFC 5280. This includes performing a bit-by-bit verification of the reference identifier. As part of the x509 certificate validation process for the HX server during TLS connectivity establishment, the TOE verifies the reference identifier found in the SAN (Subject Alternative Name) or the CN (Common Name) field. The TOE supports exact matching and wildcard usage for both types of identifiers, with the condition that the wildcard is the entire left-most label. However, it does not support IP address as reference identifier or certificate pinning. All communication between the agent and the HX server occurs exclusively over TLS, with one exception: the fast poll check, which employs non-encrypted HTTP for a basic server check. It's important to note that the fast-polling check is considered an out-of-scope feature and is not subject to evaluation. All other forms of communication, except for fast polling, inherently utilize TLS encryption, thereby ensuring mutual authentication. No configuration is required on the TOE to enable it to participate in mutual authentication during TLS communication. The TOE will automatically transmit its x509 certificate when requested by the HX server during TLS communication. If the HX server does not send the certificate message request, the client will not transmit its certificate. These TLS certificates are securely stored within the encrypted agent database.
FDP_DEC_EXT.1 FDP_NET_EXT.1	The TOE never processes or sends PII data outside the boundary of the host platform. The only external communication that is supported by the TOE is with the associated Trellix Endpoint Security (HX) Series appliance. The communication consists of a fast-polling channel on port 80 which is used to receive a Boolean value about whether there are further instructions to receive. If there are, a TLS-protected channel on Port 443 is initiated with the appliance and

Requirement	TSS Description
	instructions or updates are transferred from the appliance to the TOE via the TLS session. This channel is used to download new scanning policies. The TOE then acts on these policies (e.g., performs scans on the platform). The downloaded policies may also include instructions to send the results of the scanning to the associated appliance. In that case, the TOE again initiates a TLS-protected channel on Port 443 as before. The TOE never accesses any other host platform hardware functionality besides network connectivity. Depending on the contents of the policies the TOE receives from the associated appliance, the TOE may access the host OS syslog. The contents of the memory and the filesystem are scanned as well, leveraging the functionality provided by a kernel driver (fekern.sys) which is installed with the TOE.
FDP_DAR_EXT.1	The only sensitive information stored by the Target of Evaluation (TOE) is its RSA cryptographic private key. Additionally, the TOE retains other non-sensitive information, including security policies, its x509 certificate, the TOE identity, and the associated Trellix Endpoint Security (HX) Series appliance identity. All information, whether sensitive or not, is stored within the TOE's database. To safeguard this database from unauthorized access, it is encrypted using the Advanced Encryption Standard (AES), as outlined in FCS_STO_EXT.1
FIA_X509_EXT.1 FIA_X509_EXT.2	The TOE utilizes X.509v3 certificates, as defined in RFC 5280, to facilitate authentication for TLS connections. The validation of X.509 certificates follow the certificate path validation algorithm specified in RFC 5280, which encompasses the following checks: • Verification of the public key algorithm and parameters. • Validation of the current date and time against the certificate's validity period is performed for all TLS connections, with the exception of the initial time synchronization between the TOE and the HX server. • Verification of revocation status. • Matching the issuer name of certificate X with the subject name of certificate X+1. • Verification of name constraints. • Validation of policy Object Identifiers (OIDs). • Checking policy constraints, ensuring issuers possess CA signing capabilities. • Verification of the path length. • Processing of critical extensions. The TOE exclusively accepts Certificate Revocation List (CRL) files managed by the PKI service within the Management Console of the associated Trellix Endpoint Security (HX) Series appliance to determine the revocation status of appliance certificates. In cases where the PKI service is inactive, and the CRL is inaccessible, the TOE relies on the last known state of the HX certificate.

Requirement	TSS Description
	It's important to note that the TOE does not support pinned certificates. While wildcards are supported, they only match in the left-most label and do not match with labels featuring an explicit prefix or suffix. At every startup of the TOE, it will start a secure connection to the HX server, to synchronize its clock before doing any other tasks. Before the first-time synchronization, the TOE will not process HX server presented x509 certificates, and it can accept expired HX server certificates until the TOE's clock is synchronized. The TOE's certificate will be signed by the HX server during the provisioning phase. Throughout operation, the TOE exclusively communicates with the HX server within a closed environment. Only one certificate is assigned to the TOE for its own use, meaning it will present only this certificate in cases where validation by the HX server is required.
FMT_MEC_EXT.1	After installing the agent software, you can configure specific settings stored in the agent_config.json file. Trellix advises against manually altering many settings in the agent_config.json file. Some settings in this file should only be modified with guidance from your Trellix support representative, typically for troubleshooting purposes. The TOE's guidance document outlines common settings in the agent_config.json file and explains whether and how they can be modified. It also provides instructions for editing the agent_config.json file and ensuring its validity after any changes. The TOE stores the settings configured during installation in the C:\ProgramData\FireEye\xagt directory. Settings under the 'process' section and 'FIPS' section are among those that were set in the evaluated configuration.
FMT_CFG_EXT.1	The TOE does not use authentication for the users of the host OS. No other functionality is available until after the TOE is installed on the host platform. No modifications may be made to the TOE or its associated data by any host platform. There are no human users on the TOE. The TOE does not necessitate any form of credentials for communication with the HX server, except for authentication via an x509 certificate. The Trellix Endpoint Security (HX) Server provides a TOE's certificate, in the agent configuration file that is included in agent download packages. Changing any of x509 certificates inputs in the initial TOE's configuration file will result in the failure to install the TOE. The certificates included in the TOE's configuration file are an x509 Certificate authority certificate in addition to the TOE's certificate.
FMT_SMF.1	The TOE's administrator should have the capability to enable Microsoft Windows exploit guard protection on the TOE's platforms. This is accomplished by ensuring that the process settings section is included in the TOE's JSON configuration file.
FPR_ANO_EXT.1	The TOE does not transmit PII over the network.
FPT_API_EXT.1	The TOE leverages the following platform provided Application Programing Interfaces (APIs): – CryptAcquireContextW – CryptGenRandom – CryptReleaseContext – CryptProtectData

Requirement	TSS Description
	– CryptUnprotectData
FPT_AEX_EXT.1	The TOE never allocates memory with both write and execute permission. Write execution is always separate from execute. 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. TOE executables are written to "C:\Program Files (x86)\FireEye\xagt", in which no other files are written. In particular, no executable files are co-located in the directory in which the software is installed. During compilation the TOE is built with several flags enabled that check for engineering flaws. The flags used (or not used) are the following: – The TOE is built with the /GS flag enabled. This reduces the possibilities of stack-based buffer overflows in the product. – The compiler enables ASLR by default. – The TOE is not compiled with the /DYNAMICBASE:NO which would disable ASLR.
FPT_TUD_EXT.1	The TOE, initial installation as well as updates, is distributed as an operating system specific MSI package file. The TOE software version can be queried via the Microsoft command prompt by invoking TOE with the -v parameter. TOE updates are signed using digital certificates. The MSI packages are signed using certificates from a public trust chain which leads to DigiCert. Some components of the installation package (for instance, RemediationWSC), are signed using certificates with a public trust chain which leads to Sectigo. Updates are distributed as MSI packages provided by the associated Trellix Endpoint Security (HX) Series appliances. The TOE provides the ability to completely remove all application files when uninstalled. The user can use the platform provided web browser to query the HX series appliance to determine if an update is available. Users of the TOE should report any security related issues via the Trellix webpage (https://www.trellix.com/en-us/support/fe-support.html [trellix.com]), which provides a secure channel. Software updates/fixes are also provided via the Trellix webpage. Public availability of an update for a publicly disclosed vulnerability is typically 90 days or less and a maximum of 180 days.
FPT_TUD_EXT.2	The updates to the TOE are distributed as MSI package files. The MSI package files are signed using certificates with a public trust chain which leads to DigiCert. Some components of the installation package (for instance, RemediationWSC), are signed using certificates with a public trust chain which leads to Sectigo. Updates are distributed as MSI files provided by the associated HX appliances. The TOE provides the ability to completely remove all application files when uninstalled. The user can use the platform provided web browser to query the HX

Requirement	TSS Description
	series appliance to determine if an update is available.
FPT_LIB_EXT.1	For TOE is packaged with the following libraries: – Libuv.dll, – fips.dll – legacy.dll – libcrypto-3-x64.dll – libcrypto-3.dll – libssl-3-x64.dll – libssl-3.dll – visual studio runtime libraries (vcruntime140.dll, vccorlib140.dll, msvcp140.dll, concrt140.dll, ucrtbase.dll, api-ms-win-core-console-l1-1-0.dll, api-ms-win-core-console-l1-2-0.dll, api-ms-win-core-datetime-l1-1-0.dll, api-ms-win-core-debug-l1-1-0.dll, api-ms-win-core-errorhandling-l1-1-0.dll, api-ms-win-core-fibers-l1-1-0.dll, api-ms-win-core-file-l1-1-0.dll, api-ms-win-core-file-l1-2-0.dll, api-ms-win-core-file-l2-1-0.dll, api-ms-win-core-handle-l1-1-0.dll, api-ms-win-core-heap-l1-1-0.dll, api-ms-win-core-interlocked-l1-1-0.dll, api-ms-win-core-libraryloader-l1-1-0.dll, api-ms-win-core-localization-l1-2-0.dll, api-ms-win-core-memory-l1-1-0.dll, api-ms-win-core-namedpipe-l1-1-0.dll, api-ms-win-core-processenvironment-l1-1-0.dll, api-ms-win-core-processthreads-l1-1-0.dll, api-ms-win-core-processthreads-l1-1-1.dll, api-ms-win-core-profile-l1-1-0.dll, api-ms-win-core-rtlsupport-l1-1-0.dll, api-ms-win-core-string-l1-1-0.dll, api-ms-win-core-synch-l1-1-0.dll, api-ms-win-core-synch-l1-2-0.dll, api-ms-win-core-sysinfo-l1-1-0.dll, api-ms-win-core-timezone-l1-1-0.dll, api-ms-win-core-util-l1-1-0.dll, api-ms-win-crt-conio-l1-1-0.dll, api-ms-win-crt-convert-l1-1-0.dll, api-ms-win-crt-environment-l1-1-0.dll, api-ms-win-crt-filesystem-l1-1-0.dll, api-ms-win-crt-heap-l1-1-0.dll, api-ms-win-crt-locale-l1-1-0.dll, api-ms-win-crt-math-l1-1-0.dll, api-ms-win-crt-multibyte-l1-1-0.dll, api-ms-win-crt-private-l1-1-0.dll, api-ms-win-crt-process-l1-1-0.dll, api-ms-win-crt-runtime-l1-1-0.dll, api-ms-win-crt-stdio-l1-1-0.dll, api-ms-win-crt-string-l1-1-0.dll, api-ms-win-crt-time-l1-1-0.dll, api-ms-win-crt-utility-l1-1-0.dll, msvcp140_1.dll, msvcp140_2.dll, msvcp140_atomic_wait.dll, msvcp140_codecvt_ids.dll, vcruntime140_1.dll), – zlib1.dll
FPT_IDV_EXT.1	The TOE is distributed as a host platform specific package file providing a consistent and reliable versioning. The xAgent deployed with the .swidtag versioning file in the installation directory. After initial installation, all updates to the TOE are distributed as a package file. Each TOE installation and update is digitally signed by Trellix to ensure the authenticity of the origin. Further, the package files are downloaded from the associated Trellix Endpoint Security (HX) Series appliance over a mutually authenticated TLS session which ensures that they are always downloaded from an authentic appliance.
FTP_DIT_EXT.1	The TOE communicates externally with one trust IT entity, the Trellix Endpoint Security (HX) Series appliances. The TOE periodically polls the appliance for policy updates. To do this the TOE initiates a TLS 1.2 secured tunnel using the TOE cryptographic implementation. Updates to the scanning policies are sent through this TLS 1.2 tunnel. No additional information is sent from the TOE. All

Requirement	TSS Description
	cryptographic functionality, including the TLS protocol, is provided by the OpenSSL library, which is included within the TOE boundary.

8 Acronyms

Table 13 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
APP-PP	Application Software Protection Profile
ASLR	Address Space Layout Randomization
CA	Certification Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CM	Configuration Management
CN	Common Name
CMC	Certificate Management over CMS
CMS	Cryptographic Message Syntax
CRL	Certificate Revocation List
DRBG	Deterministic Random Bit Generation
EKU	Extended Key Usage
EP	Extension Package
FIPS	Federal Information Processing Standard
HMAC	Hashed Message Authentication Code
HTTPS	Hypertext Transfer Protocol Security
MSI	Microsoft Installer
NIAP	Nation Information Assurance Partnership
OCSP	Online Certificate Status Protocol
OID	Object Identification
OS	Operating System
OSP	Organizational Security Policy
PII	Personally Identifiable Information
PKI	Public Key Infrastructure
PP	Protection Profile
RA	Registration Authority
RAM	Random Access Memory
RFC	Request For Comments
RHEL	Red Hat Enterprise Linux
RSA	Rivest, Shamir, & Adleman
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm

Acronym	Definition
SP	[National Institute of Standards and Technology] Special Publication
ST	Security Target
TD	[NIAP] Technical Decision
TOE	Target of Evaluation
TLS	Transport Layer Security
TSS	TOE Summary Specification
UUID	Unique User Identity
VID	Verification Identification