VMware Carbon Black Endpoint Detection and Response (EDR) Windows Sensor 7.2

Security Target

ST Version: 1.5
July 21, 2021

VMware Carbon Black
1100 Winter Street
Waltham, MA 02451

Prepared By:

Booz | Allen | Hamilton
delivering results that endure

Cyber Assurance Testing Laboratory
1100 West St.
Laurel, MD 20707

Table of Contents

1 Security Target Introduction 1
1.1 ST Reference 1
1.1.1 ST Identification 1
1.1.2 Document Organization 1
1.1.3 Terminology 2
1.1.4 Acronyms 3
1.1.5 References 4
1.2 TOE Reference 4
1.3 TOE Overview 4
1.4 TOE Type 5
2 TOE Description 7
2.1 Evaluated Components of the TOE 7
2.2 Components and Applications in the Operational Environment 7
2.3 Excluded from the TOE 8
2.3.1 Not Installed 8
2.3.2 Installed but Requires a Separate License 8
2.3.3 Installed Functionality Excluded from the Evaluation 8
2.4 Physical Boundary 8
2.4.1 Hardware 8
2.4.2 Software 8
2.5 Logical Boundary 8
2.5.1 Cryptographic Support 9
2.5.2 User Data Protection 9
2.5.3 Identification and Authentication 9
2.5.4 Security Management 9
2.5.5 Privacy 9
2.5.6 Protection of the TSF 9
2.5.7 Trusted Path/Channels 10
3 Conformance Claims 11
3.1 CC Version 11
3.2 CC Part 2 Conformance Claims 11
3.3 CC Part 3 Conformance Claims 11
3.4 PP Claims 11
3.5 Package Claims 11
3.6 Package Name Conformant or Package Name Augmented 11
3.7 Conformance Claim Rationale 11
3.8 Technical Decisions 12
4 Security Problem Definition 14
4.1 Threats 14
4.2 Organizational Security Policies 14
4.3 Assumptions 14
4.4 Security Objectives 16
4.4.1 TOE Security Objectives 16
4.4.2 Security Objectives for the Operational Environment 17
4.5 Security Problem Definition Rationale 17
5 Extended Components Definition 18
5.1 Extended Security Functional Requirements 18
5.2 Extended Security Assurance Requirements 18

6 Security Functional Requirements 19
6.1 Conventions 19
6.2 Security Functional Requirements Summary 19
6.3 Security Functional Requirements 20
6.3.1 Class FCS: Cryptographic Support 20
6.3.2 Class FDP: User Data Protection 20
6.3.3 Class FMT: Security Management 21
6.3.4 Class FIA: Identification and Authentication 22
6.3.5 Class FPR: Privacy 23
6.3.6 Class FPT: Protection of the TSF 23
6.3.7 Class FTP: Trusted Path/Channels 25
6.4 Statement of Security Functional Requirements Consistency 25
7 Security Assurance Requirements 26
7.1 Class ASE: Security Target evaluation 26
7.1.1 ST introduction (ASE_INT.1) 26
7.1.2 Conformance claims (ASE_CCL.1) 27
7.1.3 Security objectives for the operational environment (ASE_OBJ.1) 28
7.1.4 Extended components definition (ASE_ECD.1) 29
7.1.5 Stated security requirements (ASE_REQ.1) 30
7.1.6 TOE summary specification (ASE_TSS.1) 31
7.2 Class ADV: Development 31
7.2.1 Basic Functional Specification (ADV_FSP.1) 31
7.3 Class AGD: Guidance Documentation 32
7.3.1 Operational User Guidance (AGD_OPE.1) 32
7.3.2 Preparative Procedures (AGD_PRE.1) 33
7.4 Class ALC: Life Cycle Support 34
7.4.1 Labeling of the TOE (ALC_CMC.1) 34
7.4.2 TOE CM Coverage (ALC_CMS.1) 34
7.4.3 Timely Security Updates (ALC_TSU_EXT.1) 35
7.5 Class ATE: Tests 35
7.5.1 Independent Testing - Conformance (ATE_IND.1) 35
7.6 Class AVA: Vulnerability Assessment 36
7.6.1 Vulnerability Survey (AVA_VAN.1) 36
8 TOE Summary Specification 37
8.1 Cryptographic Support 37
8.1.1 FCS_CKM_EXT.1 37
8.1.2 FCS_RBG_EXT.1 37
8.1.3 FCS_STO_EXT.1 37
8.2 User Data Protection 38
8.2.1 FDP_DAR_EXT.1 38
8.2.2 FDP_DEC_EXT.1 38
8.2.3 FDP_NET_EXT.1 38
8.3 Identification and Authentication 39
8.3.1 FIA_X509_EXT.1 and FIA_X509_EXT.2 39
8.4 Security Management 40
8.4.1 FMT_CFG_EXT.1 40
8.4.2 FMT_MEC_EXT.1 40
8.4.3 FMT_SMF.1 41
8.5 Privacy 41
8.5.1 FPR_ANO_EXT.1 41

8.6 Protection of the TSF ... 41
8.6.1 FPT_AEX_EXT.1 ... 41
8.6.2 FPT_API_EXT.1 ... 42
8.6.3 FPT_IDV_EXT.1 ... 43
8.6.4 FPT_LIB_EXT.1 ... 43
8.6.5 FPT_TUD_EXT.1 and FPT_TUD_EXT.2 ... 43
8.7 Trusted Path/Channels ... 60
8.7.1 FTP_DIT_EXT.1 ... 60

Table of Figures

Figure 1: TOE Boundary ... 5

Table of Tables

Table 1: Customer Specific Terminology ... 2
Table 2: CC Specific Terminology ... 3
Table 3: Acronym Definition ... 3
Table 4: Evaluated Components of the TOE ... 7
Table 5: Components of the Operational Environment ... 7
Table 6: Technical Decisions ... 14
Table 7: TOE Threats ... 14
Table 8: TOE Assumptions ... 15
Table 9: TOE Security Objectives ... 16
Table 10: Operational Environment Objectives ... 17
Table 11: Security Functional Requirements for the TOE ... 20
Table 12: Certificate and Key Store ... 38
Table 13: API listing ... 42

1 Security Target Introduction

This chapter presents the Security Target (ST) identification information and an overview. An ST contains the Information Technology (IT) security requirements of an identified Target of Evaluation (TOE) and specifies the functional and assurance security measures offered by the TOE.

1.1 ST Reference

This section provides information needed to identify and control this ST and its Target of Evaluation.

1.1.1 ST Identification

ST Title:	VMware Carbon Black Endpoint Detection and Response (EDR) Windows Sensor 7.2 Security Target
ST Version:	1.5
ST Publication Date:	July 21, 2021
ST Author:	Booz Allen Hamilton

1.1.2 Document Organization

Chapter 1 of this document provides identifying information for the ST and TOE as well as a brief description of the TOE and its associated TOE type.

Chapter 2 describes the TOE in terms of its physical boundary, logical boundary, exclusions, and dependent Operational Environment components.

Chapter 3 describes the conformance claims made by this ST.

Chapter 4 describes the threats, assumptions, objectives, and organizational security policies that apply to the TOE.

Chapter 5 defines extended Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs).

Chapter 6 describes the SFRs that are to be implemented by the TSF.

Chapter 7 describes the SARs that will be used to evaluate the TOE.

Chapter 8 provides the TOE Summary Specification, which describes how the SFRs that are defined for the TOE are implemented by the TSF.

1.1.3 Terminology

This section defines the terminology used throughout this ST. The terminology used throughout this ST is defined in Table 1 and 2. These tables are to be used by the reader as a quick reference guide for terminology definitions.

Term	Definition
Local Administrator	The Windows OS administrator who has system permissions to access sensitive data and perform management functionality on the endpoint system.
Management Server	Operational environment server that is used for the remote management of VMware CB EDR Windows Sensors (TOE) by an enterprise administrator. This is a separate product called VMware Carbon Black EDR Server (VMware CB EDR Server) and is not part of the TOE boundary.
Endpoint System	A device or set of devices, such as a laptop or desktop, with the Windows operating system that hosts the TOE.
Endpoint User	An individual who has access to the TOE but is not able to manage its behavior.
Sensor Group	Each host sensor is associated with a sensor group that defines its configuration and security characteristics. A sensor group must contain at least one host sensor and can contain many host sensors. However, a single host sensor can only belong to one sensor group. Sensor groups can be based on the security and organizational requirements. For example, one could base sensor groups on functional groupings/departments (such as marketing, customer service, or IT) or location.

Table 1: Customer Specific Terminology

Term	Definition
Address Space Layout Randomization (ASLR)	An anti-exploitation feature which loads memory mappings into unpredictable locations. ASLR makes it more difficult for an attacker to redirect control to code that they have introduced into the address space of an application process.
Application (app)	Software that runs on a platform and performs tasks on behalf of the user or owner of the platform, as well as its supporting documentation. The terms *TOE* and *application* are interchangeable in this document.
Application Programming Interface (API)	A specification of routines, data structures, object classes, and variables that allows an application to make use of services provided by another software component, such as a library. APIs are often provided for a set of libraries included with the platform.
Credential	Data that establishes the identity of a user, e.g. a cryptographic key or password.
Data Execution Prevention (DEP)	An anti-exploitation feature of modern operating systems executing on modern computer hardware, which enforces a non-execute permission on pages of memory. DEP prevents pages of memory from containing both data and instructions, which makes it more difficult for an attacker to introduce and execute code.
Developer	An entity that writes application software. For the purposes of this document, vendors and developers are the same.
Operating System (OS)	Software that manages hardware resources and provides services for applications.
Personally Identifiable Information (PII)	Any information about an individual maintained by an agency, including, but not limited to, education, financial transactions, medical history, and criminal or employment history and information which can be used to distinguish or trace an individual's identity, such as their name, social security number, date and place of birth, mother's maiden

	name, biometric records, etc., including any other personal information which is linked or linkable to an individual.
Platform	The environment in which application software runs. The platform can be an operating system, hardware environment, a software based execution environment, or some combination of these. These types platforms may also run atop other platforms.
Sensitive Data	Sensitive data may include all user or enterprise data or may be specific application data such as emails, messaging, documents, calendar items, and contacts. Sensitive data must minimally include PII, credentials, and keys. Sensitive data shall be identified in the TSS by the ST author.
Trusted Channel	An encrypted connection between the TOE's host platform and a system in the Operational Environment.
Trusted Path	An encrypted connection between the TOE and the application an Authorized Administrator uses to manage it (web browser, terminal client, etc.).
User	In a CC context, any individual who has the ability to manage TOE functions or data.
Vendor	An entity that sells application software. For purposes of this document, vendors and developers are the same. Vendors are responsible for maintaining and updating application software.

Table 2: CC Specific Terminology

1.1.4 Acronyms

The acronyms used throughout this ST are defined in Table 3. This table is to be used by the reader as a quick reference guide for acronym definitions.

Acronym	Definition
API	Application Programming Interface
ASLR	Address Space Layout Randomization
CA	Certificate Authority
CB EDR	Carbon Black Endpoint Detection and Response
CC	Common Criteria
CLI	Command Line Interface
DEP	Data Execution Prevention
HTTPS	Hyper Text Transfer Protocol Secure
IT	Information Technology
NIAP	National Information Assurance Partnership
OS	Operating System
PII	Personally Identifiable Information
PP	Protection Profile
DRBG	Deterministic Random Bit Generator
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function

Table 3: Acronym Definition

1.1.5 References

[1] Protection Profile for Application Software, version 1.3 [APP_PP]

[2] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-001

[3] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-002

[4] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-003

[5] Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-004 [CEM]

[6] NIST Special Publication 800-56A Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography Rev. 3 April 2018

[7] FIPS PUB 186-4 Federal Information Processing Standards Publication Digital Signature Standard July 2013

[8] Microsoft Windows Common Criteria Evaluation Microsoft Windows 10 version 1903 (May 2019 Update) Microsoft Windows Server version 1903 (May 2019 Update) Security Target Version 0.04 July 19, 2019

[9] VMware Carbon Black Endpoint Detection and Response (EDR) Windows Sensor 7.2 Supplemental Administrative Guidance for Common Criteria, June 17, 2020

1.2 TOE Reference

The TOE is the VMware Carbon Black Endpoint Detection and Response (EDR) Windows Sensor 7.2 (VMware CB EDR Windows Sensor), which is an application executing on an operating system.

1.3 TOE Overview

The TOE is the VMware Carbon Black Endpoint Detection and Response (EDR) Windows Sensor 7.2 application, also referred to as the TOE from this point forward. The VMware CB EDR Windows Sensor is an enterprise software application whose primary purpose is to gather event data on the endpoints and invoke the OS to securely transmit this information to the operating environment's management server for centralized storage and indexing.

The TOE is installed on administratively defined network endpoints, such as laptops, desktops, and servers. The TOE, when installed, operates as a Windows service to perform its function of observing and reporting on system-level behavior. Changes to the TOE's data collection policy can only be initiated by the enterprise administrator using the management server in the operational environment.

In the evaluated configuration, as depicted in Figure 1, the TOE's evaluation scope is only the VMware CB EDR Windows Sensor application and its configuration information. The TOE operates on top of Microsoft Windows 10 and Windows Server 2019 OS with BitLocker (or equivalent) enabled. The data collection policy configuration settings are stored in the Windows registry (not shown). Additionally, the TOE stores the collected event data into its own data storage on the local file system until transferred to the management server using a HTTPS session over TLS v1.2 (HTTPS/TLS) trusted channel. All cryptographic functionality used to secure remote communications is provided by the underlying Windows OS.

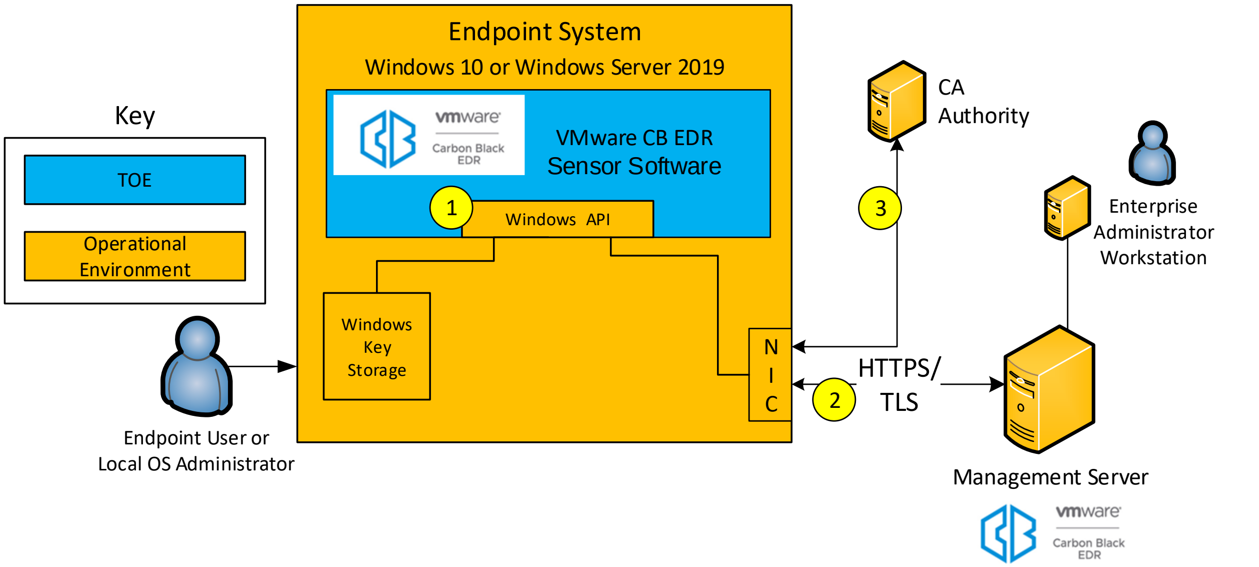
Endpoint System
Windows 10 or Windows Server 2019
VMware CB EDR Sensor Software
Windows API
Windows Key Storage
N I C
Key
TOE
Operational Environment
Endpoint User or Local OS Administrator
CA Authority
Enterprise Administrator Workstation
HTTPS/ TLS
Management Server
1
2
3

Figure 1: VMware Carbon Black EDR Windows Sensor TOE Boundary

The TOE is automatically executed by the OS upon start or reboot of the host platform. This same executable can be used by the local OS administrator to verify the version of the TOE. All other management functions such as starting, stopping, and performing a forced uninstallation of the TOE is performed by the local OS administrator using Window interfaces. The TOE application does not provide any interactive user interface that allows an endpoint user to manage the TOE locally or provide a means to generate and store user-modifiable data.

Supporting Environment Interfaces:

- **TOE Application to OS (1)** –The TOE leverages operating system callbacks in order to collect system-relevant data information, store log files, access Windows key store, and invoke network access.
- **TOE Platform to Management Server (2)** – The TOE invokes the operating system to use HTTPS/TLS to establish a trusted communication channel to the management server for transmitting the collected data files, retrieving configuration updates, and obtaining software updates. The TOE platform (OS) is the HTTPS/TLS client for this interface and performs X.509 certificate validation in support of the non-mutually authenticated HTTPS/TLS communications.
- **TOE Platform to CA Authority (3)** – The TOE platform (OS) performs X.509 certificate validation in support of the non-mutually authenticated HTTPS/TLS communications to the Management Server.

1.4 TOE Type

The TOE is application software that is deployed on individual endpoint systems running a Windows operating system. The [APP_PP] states the following:

“The application, which consists of the software provided by its vendor, is installed onto the platform(s) it operates on. It executes on the platform, which may be an operating system, hardware environment, a software based execution environment, or some combination of these. Those platforms may themselves run within other environments, such as virtual machines or operating systems, that completely abstract away the underlying hardware from the application. The TOE is not accountable for security functionality that is implemented by platform layers that are abstracted away. Some evaluation activities are specific to the particular platform on which the application runs, in order to provide precision and repeatability."

The Application Software TOE type is justified because:

- the TOE is application software that must be installed onto the platform it will operates on
- the TOE executes on a hardware platform that is running Windows 10 or Windows 2019 server
- the TOE is not accountable for security functionality provided by the platform layers.

2 TOE Description

This section provides a description of the TOE in its evaluated configuration. This includes the physical and logical boundaries of the TOE.

2.1 Evaluated Components of the TOE

The following table describes the TOE components in the evaluated configuration:

Component	Definition
VMware CB EDR Windows Sensor (TOE)	An application that is installed on a Windows 10 platform, which collects event data from this host endpoint platform and invokes the OS to securely transmit this collected information back to the management server in the Operating Environment. The TOE maintains the configuration settings in the Windows registry and on the local file system. The TOE does not provide an interactive user interface for creating or storing data on the endpoint system.

Table 4: Evaluated Components of the TOE

2.2 Components and Applications in the Operational Environment

The following table lists components and applications in the environment that the TOE relies upon in order to function properly:

Component	Definition
Endpoint system with Microsoft Windows 10 (Windows) using Intel Core I5-8350U processor.	The host platform along with the operating system installed that the TOE application is installed on.**
Management Server platform with the VMware CB EDR Server* software installed *evaluated independently	The management server is used in the evaluated configuration to deploy the TOE, collect the system data from these sensors, perform configuration updates, and deploy software updates. However, it is used to the extent that it can assist in the evaluation of the TOE software and no security claims for its functionality are made in this evaluation.
Administration Workstation	Any general-purpose computer that is used by an enterprise administrator to operate the Management Server remotely via a web browser.
Certificate Authority	The server deployed within the Operational Environment which confirms the validity and revocation status of certificates. This is only required for the TOE to validate TOE server certificate.

Table 5: Components of the Operational Environment

**NOTE: It is expected that the TOE is operating on a Common Criteria certified operating system and platform based on the Microsoft Windows 10 and Server 2019 version 1903 (May 2019 Update) evaluation. The TOE software that is installed on the Windows based OS, is identical (installation, executables, functionality, and features) no matter which variation of the Windows 10 (May 2019 Update) is used. For testing, the TOE was installed and fully tested on the Windows 10 Enterprise 2019 variant.

2.3 Excluded from the TOE

The following optional products, components, and/or applications can be integrated with the TOE but are not included in the evaluated configuration. They provide no PP claimed security related functionality for the evaluated product. They are separated into three categories: not installed, installed but requires a separate license, and installed but not part of the TSF.

2.3.1 Not Installed

There are no components that are not installed.

2.3.2 Installed but Requires a Separate License

There are no excluded components that are installed and require a separate license.

2.3.3 Installed Functionality Excluded from the Evaluation

The TOE only includes the functionality that satisfies the Security Functional Requirements (SFRs) in the claimed Protection Profile. Therefore, the following product functionality is considered out of scope as there are no SFRs that can be mapped to this functionality:

- The functionality of collecting event data on the TOE. This includes the use of the following executables, which are tools that are executed by the main program (cb.exe):
  - *sensordiag.exe*
  - *osqueryi.exe*
- The functionality of enforcing incident response actions dictated by the management server administrator: This includes the use of the following executables, which are tools that are executed by the main program executable (cb.exe):
  - *cbmarshal.exe*
  - *cbedrcli.exe*

2.4 Physical Boundary

2.4.1 Hardware

This is a software-only TOE. All hardware that is present is part of the TOE’s Operational Environment.

2.4.2 Software

The physical boundary of the TOE software is the VMware CB EDR Windows Sensor application and its configuration data.

2.5 Logical Boundary

The TOE is comprised of several security features. Each of these security features consists of several security functionalities, as identified below. This ST includes the security functional requirements from the Application Software Protection Profile v1.3.

1. Cryptographic Support
2. User Data Protection

3. Security Management
4. Privacy
5. Protection of the TSF
6. Trusted Path/Channels

2.5.1 Cryptographic Support

The TOE invokes the underlying platform to perform all cryptographic services including HTTPS/TLS trusted communications, and sensitive data encryption storage. As an application on an operating system, the TOE interfaces with the operating system's key storage to securely store key data related to secure communications.

2.5.2 User Data Protection

The application restricts its access to the endpoint system's network connectivity resources. It also restricts its sensitive data access to system logs and memory dumps stored on the endpoint system. Network activity is restricted to periodic management server polling, aka sensor check-in. During the periodic polling, the Sensor transmits sensor collected endpoint system data to the management server, retrieves configuration settings/updates and TOE software updates (if available) from the management server.

2.5.3 Identification and Authentication

The TOE relies on the OS to validate X.509.3 certificates for HTTPS/TLS communication.

2.5.4 Security Management

During installation, the TOE is automatically configured to protect itself and its data from unauthorized access and implements the recommended Windows platform security mechanisms. The TOE application provides one CLI that provides the ability for an OS administrator to verify the application version. The TSF implements changes to its configuration received during the polling cycle from the management server.

2.5.5 Privacy

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

2.5.6 Protection of the TSF

The TOE is packaged as separate software that is installed on the platform and can be uninstalled/removed if needed. In the evaluated configuration, all updates are obtained from the management server. The digital signature of the update package is verified by the host platform prior to being installed. The TOE will only initiate an update when the management server has indicated, during the periodic polling cycle, there is an authorized update available. Otherwise the TOE does not download, replace, or modify its own binary code.

The TOE implements anti-exploitation features, such as stack-based overflow protection, is compatible with security features provided by the OS, and only uses documented APIs and libraries.

2.5.7 Trusted Path/Channels

The TOE invokes the OS platform to provide a trusted communication channel (HTTPS session over TLS v1.2) to the management server.

3 Conformance Claims

3.1 CC Version

This ST is compliant with Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 5 April 2017.

3.2 CC Part 2 Conformance Claims

This ST and Target of Evaluation (TOE) is Part 2 extended to include all applicable NIAP and International interpretations through July 21, 2021.

3.3 CC Part 3 Conformance Claims

This ST and Target of Evaluation (TOE) are conformant to Part 3 extended to include all applicable NIAP and International interpretations through July 21, 2021.

3.4 PP Claims

This ST claims exact conformance to the following Protection Profiles:

- Protection Profile for Application Software Version 1.3 [APP_PP]

3.5 Package Claims

The TOE claims exact compliance to the *Protection Profile for Application Software*

The TOE claims the following selection-based SFR that is defined in the appendices of the claimed APP_PP:

- FPT_TUD_EXT.2

This does not violate the notion of exact conformance because the PP specifically indicates these as allowable options and provides both the ST author and evaluation laboratory with instructions on how these claims are to be documented and evaluated.

3.6 Package Name Conformant or Package Name Augmented

This ST and TOE claim exact conformance to the [APP_PP].

3.7 Conformance Claim Rationale

The [APP_PP] states the following:

“The App PP states the following: “The requirements in this document apply to application software which runs on any type of platform. Some application types are covered by more specific PPs, which may be expressed as PP-Modules of this PP. Such applications are subject to the requirements of both this PP and the PP-Module that addresses their special functionality. PPs for some particularly specialized applications may not be expressed as PP-Modules at this time, though the requirements in this document should be seen as objectives for those highly specialized applications.

Although the requirements in this document apply to a wide range of application software, consult guidance from the relevant national schemes to determine when formal Common Criteria evaluation is expected for a particular type of application. This may vary depending upon the nature of the security functionality of the application."

The TOE is a standalone application which runs on a Windows OS platform and is therefore considered to be relevant to the App PP. There are no PP-Modules to the App PP that are applicable to the product, so the TOE is characterized only as a software application.

3.8 Technical Decisions

The evaluator used the following methodology for analyzing the technical decisions:

If the TD is a modification of wording to an SFR that is being claimed by the TOE then the Analysis will indicate 'Applicable' and any further relevant details such as"

- 'changes applied' and the Technical Decisions SFR wording updates have been applied and annotated with a Footnote.
- 'not selecting modified option for …' to indicate reader won't find the wording changes identified by TD
- 'AA: TSS, AGD, Test' to indicate Assurance Activity change specific to TSS, AGD, or Testing

If the TD is a modification of the wording to an SFR or an Assurance Activity for an SFR that is not being claimed by the TOE, then the Analysis NA column will be filled in with an 'X' with the reason "not claiming…."

The following is a complete list of Technical Decisions that apply to the [APP_PP] evaluation activities that must be considered for the evaluation of this TOE.

TD #	Title	References	Changes			Analysis to this evaluation	
			SFR	**AA**	**Notes**	**NA**	**Reason**
TD0587	X.509 SFR Applicability in App PP	FIA_X509_EXT.1, FIA_X509_EXT.2, FTP_DIT_EXT.1	X	X	X		FIA_X509_EXT.2.1 has additional SFR option. However, this option not selected. Therefore, no footnote or identification is reflected in ST. FIA_X509_EXT1.1 SFR wording was updated. Footnote 3 Additionally, the AA: TSS, Test wording FTP_DIT_EXT.1 has additional SFR option However, this option not selected.

							Therefore, no footnote or identification is reflected in ST.
TD0582	PP-Configuration for Application Software and Virtual Private Network (VPN) Clients now allowed	FDP_DAR_EXT,1	X		X		Update to conformance claims. Corrected SFR option. However, this option not selected. Therefore, no footnote or identification is reflected in ST.
TD0561	Signature verification update	FPT_TUD_EXT.1.4, FPT_TUD_EXT.2	X	X			AA:TSS Footnotes: 4 and 5
TD0554	iOS/iPadOS/Android AppSW Virus Scan	AVA_VAN.1		X			AA: Test modified for iOS/Android platforms and update for AVA_VAN search and analysis applicable.
TD0548	Integrity for installation tests in AppSW PP 1.3	FPT_TUD_EXT.1.3		X			AA: Updated Test for iOS; Test wording applied to all other platforms
TD0544	Alternative testing methods for FPT_AEX_EXT.1.1	FPT_AEX_EXT.1		X			AA: Test wording applied to all platforms
TD0543	FMT_MEC_EXT.1 evaluation activity update	FMT_MEC_EXT.1		X			AA: Test modified for Windows. Not claiming .NET or WUA
TD0540	Expanded AES Modes in FCS_COP	FCP_COP.1(1)	X	X		X	Not claiming Cryptography. All handled by OS. SFR modified; AES-CCM Tests modified
TD0519	Linux symbolic links and FMT_CFG_EXT.1	FMT_CFG_EXT.1.2		X		X	AA: Test modified for Linux
TD0515	Use Android APK manifest in test	FDP_DEC_EXT.1		X		X	Not claiming Android
TD0510	Obtaining random bytes for iOS/macOS	FCS_RBG_EXT.1		X		X	Not claiming iOS
TD0498	Application Software PP Security Objectives and Requirements Rationale	Section 4.3 and Section 5.2 in PP			X		Updates PP rationale
TD0495	FIA_X509_EXT.1.2 Test Clarification	FIA_X509_EXT.1.2		X			
TD0473	Support for Client or Server TOEs in FCS_HTTPS_EXT	FCS_HTTPS_EXT.1, FCS_HTTPS_EXT.2	X	X	X	X	Not claiming HTTPS SFR
TD0465	Configuration Storage for .NET Apps	FMT_MEC_EXT.1		X			AA: Test However, not claiming .NET framework
TD0445	User Modifiable File Definition	FPT_AEX_EXT.1.4		X	X		AA: Test User modifiable file definition clarity
TD0437	Supported Configuration Mechanism	FMT_MEC_EXT.1.1	X	X	X		AA: TSS, Tests

							However, support of file encryption option is not claimed Footnote 2
TD0435	Alternative to SELinux for FPT_AEX_EXT.1.3	FPT_AEX_EXT.1.3		X		X	Not claiming Linux
TD0434	Windows Desktop Applications Test	FDP_DEC_EXT.1.1		X			AA Test
TD0427	Reliable Time Source	A.Platform	X				Changes to wording in ST: Updated wording to Assumption Footnote 1
TD0416	Correction to FCS_RBG_EXT.1 Test Activity	FCS_RBG_EXT.1.1		X			AA: Test wording modified for invoking the platform

Table 6: Technical Decisions

4 Security Problem Definition

4.1 Threats

This section identifies the threats against the TOE. These threats have been taken from the [APP_PP].

Threat	Threat Definition
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.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.PHYSICAL_ACCESS	An attacker may try to access sensitive data at rest.

Table 7: TOE Threats

4.2 Organizational Security Policies

There are no Organizational Security Policies in the [APP_PP].

4.3 Assumptions

The specific conditions listed in this section are assumed to exist in the TOE’s Operational Environment. These assumptions have been taken from the [APP_PP].

Assumption	Assumption Definition

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

Table 8: TOE Assumptions

[1] TD0427

4.4 Security Objectives

This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

4.4.1 TOE Security Objectives

This section identifies the security objectives of the TOE. These objectives have been taken directly from the [APP_PP].

Objective	Objective Definition
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.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.
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.
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.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.

Table 9: TOE Security Objectives

4.4.2 Security Objectives for the Operational Environment

The TOE's operational environment must satisfy the following objectives:

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

Table 10: Operational Environment Objectives

4.5 Security Problem Definition Rationale

The assumptions, threats, OSPs, and objectives that are defined in this ST represent the assumptions, threats, OSPs, and objectives that are specified in the Protection Profiles to which the TOE claims conformance. The associated mappings of assumptions to environmental objectives, SFRs to TOE objectives, and OSPs and objectives to threats are therefore identical to the mappings that are specified in the claimed Protection Profiles.

5 Extended Components Definition

5.1 Extended Security Functional Requirements

The extended Security Functional Requirements that are claimed in this ST are taken directly from the PPs to which the ST and TOE claim conformance. These extended components are formally defined in the PPs in which their usage is required.

5.2 Extended Security Assurance Requirements

The extended Security Assurance Requirement that is claimed in this ST is taken directly from the PP to which the ST and TOE claim conformance. This extended component is formally defined in the PP in which its usage is required.

6 Security Functional Requirements

6.1 Conventions

The CC permits four functional component operations—assignment, refinement, selection, and iteration—to be performed on functional requirements. This ST will highlight the operations in the following manner:

- **Assignment:** allows the specification of an identified parameter. Indicated with *italicized text*.
- **Refinement:** allows the addition of details. Indicated with **bold text**.
- **Selection:** allows the specification of one or more elements from a list. Indicated with underlined text.
- **Iteration operation:** are identified with a number inside parentheses (e.g. "(1)")

When multiple operations are combined, such as an assignment that is provided as an option within a selection or refinement, a combination of the text formatting is used.

Text that is formatted in a claimed PP, such as if the PP's instantiation of the SFR has a refinement (bolded font), or a completed assignment (inside brackets), the formatting is not preserved when reproduced in this ST. Only the assignments and selections made by the ST author are within [brackets]. This is so that the reader can easily identify the operations that are performed by the ST author.

6.2 Security Functional Requirements Summary

The following table lists the SFRs that originate from the Application Software Protection Profile [APP_PP] and are claimed by the TOE.

Class Name	Component Identification	Component Name
Cryptographic Support	FCS_CKM_EXT.1	Cryptographic Key Generation Services
	FCS_RBG_EXT.1	Random Bit Generation Services
	FCS_STO_EXT.1	Storage of Credentials
User Data Protection	FDP_DAR_EXT.1	Encryption of Sensitive Application Data
	FDP_DEC_EXT.1	Access to Platform Resources
	FDP_NET_EXT.1	Network Communications
Identification and Authentication	FIA_X509_EXT.1	X.509 Authentication and Encryption
	FIA_X509_EXT.2	X.509 Authentication and Encryption
Security Management	FMT_CFG_EXT.1	Secure by Default Configuration
	FMT_MEC_EXT.1	Supported Configuration Mechanism
	FMT_SMF.1	Specification of Management Functions
Privacy	FPR_ANO_EXT.1	User Consent for Transmission of Personally Identifiable Information
Protection of the TSF	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
	FPT_TUD_EXT.1	Integrity for Installation and Update
	FPT_TUD_EXT.2	Integrity for Installation and Update
Trusted Path/Channels	FTP_DIT_EXT.1	Protection of Data in Transit

Table 11: Security Functional Requirements for the TOE

6.3 Security Functional Requirements

6.3.1 Class FCS: Cryptographic Support

6.3.1.1 *FCS_CKM_EXT.1* *Cryptographic Key Generation Services*

FCS_CKM_EXT.1.1

The application shall [

- generate no asymmetric cryptographic keys

].

6.3.1.2 *FCS_RBG_EXT.1* *Random Bit Generation Services*

FCS_RBG_EXT.1.1

The application shall [

- use no DRBG functionality

] for its cryptographic operations.

6.3.1.3 *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 [*management server certificate chain*, *Sensor Group certificate*]

] to non-volatile memory.

6.3.2 Class FDP: User Data Protection

6.3.2.1 *FDP_DAR_EXT.1* *Encryption of Sensitive Application Data*

FDP_DAR_EXT.1

The application shall [

- leverage platform-provided functionality to encrypt sensitive data,
- protect sensitive data in accordance with FCS_STO_EXT.1

] in non-volatile memory.

6.3.2.2 *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,
- [*memory dumps*]

].

6.3.2.3 *FDP_NET_EXT.1* *Network Communications*

FDP_NET_EXT.1.1

The application shall restrict network communication to [

- [*application initiated periodic management server polling*]

].

6.3.3 Class FMT: Security Management

6.3.3.1 *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 user.

6.3.3.2 *FMT_MEC_EXT.1* *Supported Configuration Mechanism*

FMT_MEC_EXT.1.1[2]

The application shall [

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

6.3.3.3 *FMT_SMF.1* *Specification of Management Functions*

FMT_SMF.1.1

[2] TD0437

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

- enable/disable the transmission of any information describing the system's hardware, software, or configuration,
- [*Version check*]

].

6.3.4 Class FIA: Identification and Authentication

6.3.4.1 *[APP_PP] FIA_X509_EXT.1 X.509 Certificate Validation*

FIA_X509_EXT.1.1[3]

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, 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 [OCSP as specified in RFC 6960]
- 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 only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

[3] TD0587

6.3.4.2 *[APP_PP] 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].

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].

6.3.5 Class FPR: Privacy

6.3.5.1 *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

].

6.3.6 Class FPT: Protection of the TSF

6.3.6.1 *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.

6.3.6.2 *FPT_API_EXT.1* *Use of Supported Services and APIs*

FPT_API_EXT.1.1

The application shall use only documented platform APIs.

6.3.6.3 *FPT_IDV_EXT.1* *Software Identification and Versions*

FPT_IDV_EXT.1.1

The application shall be versioned with [[*major.minor.patch.build methodology*]].

6.3.6.4 *FPT_LIB_EXT.1* *Use of Third Party Libraries*

FPT_LIB_EXT.1.1

The application shall be packaged with only [*Google protobuf, zlib*].

6.3.6.5 *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 [leverage the platform] 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[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].

6.3.6.6 *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.

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.

[4] TD0561

FPT_TUD_EXT.2.3[5]

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

6.3.7 Class FTP: Trusted Path/Channels

6.3.7.1 *FTP_DIT_EXT.1* *Protection of Data in Transit*

FTP_DIT_EXT.1.1

The application shall [

- invoke platform-provided functionality to encrypt all transmitted data with [HTTPS]

] between itself and another trusted IT product.

6.4 Statement of Security Functional Requirements Consistency

The Security Functional Requirements included in the ST represent all required SFRs specified in the PPs against which exact conformance is claimed a subset of the optional SFRs. All hierarchical relationships, dependencies, and unfulfilled dependency rationales in the ST are considered to be identical to those that are defined in the claimed PP.

[5] TD0561

7 Security Assurance Requirements

This section identifies the Security Assurance Requirements (SARs) that are claimed for the TOE. The SARs which are claimed are in exact conformance with the [APP_PP].

Security Target (ASE)	ST introduction (ASE_INT.1)
	Conformance claims (ASE_CCL.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Extended components definition (ASE_ECD.1)
	Stated security requirements (ASE_REQ.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance Documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life Cycle Support (ALC)	Labelling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
Tests (ATE)	Independent testing – conformance (ATE_IND.1)
Vulnerability Assessment (AVA)	Vulnerability survey (AVA_VAN.1)

7.1 Class ASE: Security Target evaluation

7.1.1 ST introduction (ASE_INT.1)

7.1.1.1 *Developer action elements:*

ASE_INT.1.1D

The developer shall provide an ST introduction.

7.1.1.2 *Content and presentation elements:*

ASE_INT.1.1C

The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C

The ST reference shall uniquely identify the ST.

ASE_INT.1.3C

The TOE reference shall uniquely identify the TOE.

ASE_INT.1.4C

The TOE overview shall summarise the usage and major security features of the TOE.

ASE_INT.1.5C

The TOE overview shall identify the TOE type.

ASE_INT.1.6C

The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C

The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C

The TOE description shall describe the logical scope of the TOE.

7.1.1.3 *Evaluator action elements:*

ASE_INT.1.1E

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

ASE_INT.1.2E

The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

7.1.2 Conformance claims (ASE_CCL.1)

7.1.2.1 *Developer action elements:*

ASE_CCL.1.1D

The developer shall provide a conformance claim.

ASE_CCL.1.2D

The developer shall provide a conformance claim rationale

7.1.2.2 *Content and presentation elements:*

ASE_CCL.1.1C

The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C

The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C

The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C

The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C

The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C

The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C

The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C

The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C

The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C

The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

7.1.2.3 *Evaluator action elements:*

ASE_CCL.1.1E

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

7.1.3 Security objectives for the operational environment (ASE_OBJ.1)

7.1.3.1 *Developer action elements:*

ASE_OBJ.1.1D

The developer shall provide a statement of security objectives.

7.1.3.2 *Content and presentation elements:*

ASE_OBJ.1.1C

The statement of security objectives shall describe the security objectives for the operational environment.

7.1.3.3 *Evaluator action elements:*

ASE_OBJ.1.1E

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

7.1.4 Extended components definition (ASE_ECD.1)

7.1.4.1 *Developer action elements:*

ASE_ECD.1.1D

The developer shall provide a statement of security requirements.

ASE_ECD.1.2D

The developer shall provide an extended components definition.

7.1.4.2 *Content and presentation elements:*

ASE_ECD.1.1C

The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C

The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C

The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C

The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C

The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

7.1.4.3 *Evaluator action elements:*

ASE_ECD.1.1E

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

ASE_ECD.1.2E

The evaluator shall confirm that no extended component can be clearly expressed using existing components.

7.1.5 Stated security requirements (ASE_REQ.1)

7.1.5.1 *Developer action elements:*

ASE_REQ.1.1D

The developer shall provide a statement of security requirements.

ASE_REQ.1.2D

The developer shall provide a security requirements rationale.

7.1.5.2 *Content and presentation elements:*

ASE_REQ.1.1C

The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.1.2C

All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.1.3C

The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.1.4C

All operations shall be performed correctly.

ASE_REQ.1.5C

Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.1.6C

The statement of security requirements shall be internally consistent.

7.1.5.3 *Evaluator action elements:*

ASE_REQ.1.1E

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

7.1.6 TOE summary specification (ASE_TSS.1)

7.1.6.1 *Developer action elements:*

ASE_TSS.1.1D

The developer shall provide a TOE summary specification.

7.1.6.2 *Content and presentation elements:*

ASE_TSS.1.1C

The TOE summary specification shall describe how the TOE meets each SFR.

7.1.6.3 *Evaluator action elements:*

ASE_TSS.1.1E

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

ASE_TSS.1.2E

The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

7.2 Class ADV: Development

7.2.1 Basic Functional Specification (ADV_FSP.1)

7.2.1.1 *Developer action elements:*

ADV_FSP.1.1D

The developer shall provide a functional specification.

ADV_FSP.1.2D

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

7.2.1.2 *Content and presentation elements:*

ADV_FSP.1.1C

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

ADV_FSP.1.2C

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

ADV_FSP.1.3C

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

ADV_FSP.1.4C

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

7.2.1.3 *Evaluator action elements:*

ADV_ FSP.1.1E

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

ADV_ FSP.1.2E

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

7.3 Class AGD: Guidance Documentation

7.3.1 Operational User Guidance (AGD_OPE.1)

7.3.1.1 *Developer action elements:*

AGD_OPE.1.1D

The developer shall provide operational user guidance.

7.3.1.2 *Content and presentation elements:*

AGD_OPE.1.1C

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

AGD_OPE.1.2C

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

AGD_OPE.1.3C

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

AGD_OPE.1.4C

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

AGD_OPE.1.5C

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

AGD_OPE.1.6C

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

AGD_OPE.1.7C

The operational user guidance shall be clear and reasonable.

7.3.1.3 *Evaluator action elements:*

AGD_OPE.1.1E

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

7.3.2 Preparative Procedures (AGD_PRE.1)

7.3.2.1 *Developer action elements:*

AGD_PRE.1.1D

The developer shall provide the TOE including its preparative procedures.

7.3.2.2 *Content and presentation elements:*

AGD_ PRE.1.1C

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

AGD_ PRE.1.2C

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

7.3.2.3 *Evaluator action elements:*

AGD_ PRE.1.1E

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

AGD_ PRE.1.2E

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

7.4 Class ALC: Life Cycle Support

7.4.1 Labeling of the TOE (ALC_CMC.1)

7.4.1.1 *Developer action elements:*

ALC_CMC.1.1D

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

7.4.1.2 *Content and presentation elements:*

ALC_CMC.1.1C

The TOE shall be labeled with its unique reference.

7.4.1.3 *Evaluator action elements:*

ALC_CMC.1.1E

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

7.4.2 TOE CM Coverage (ALC_CMS.1)

7.4.2.1 *Developer action elements:*

ALC_CMS.1.1D

The developer shall provide a configuration list for the TOE.

7.4.2.2 *Content and presentation elements:*

ALC_CMS.1.1C

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

ALC_CMS.1.2C

The configuration list shall uniquely identify the configuration items.

7.4.2.3 *Evaluator action elements:*

ALC_CMS.1.1E

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

7.4.3 Timely Security Updates (ALC_TSU_EXT.1)

7.4.3.1 *Developer Actions Element:*

ALC_TSU_EXT.1.1D

The developer shall provide a description in the TSS of how timely security updates are made to the TOE.

ALC_TSU_EXT.1.2D

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

7.4.3.2 *Content and presentation elements:*

ALC_TSU_EXT.1.1C

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

ALC_TSU_EXT.1.1C

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

ALC_TSU_EXT.1.1C

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

7.4.3.3 *Evaluator action elements:*

ALC_TSU_EXT.1.1E

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

7.5 Class ATE: Tests

7.5.1 Independent Testing - Conformance (ATE_IND.1)

7.5.1.1 *Developer action elements:*

ATE_IND.1.1D

The developer shall provide the TOE for testing.

7.5.1.2 *Content and presentation elements:*

ATE_IND.1.1C

The TOE shall be suitable for testing.

7.5.1.3 *Evaluator action elements:*

ATE_IND.1.1E

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

ATE_IND.1.2E

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

7.6 Class AVA: Vulnerability Assessment

7.6.1 Vulnerability Survey (AVA_VAN.1)

7.6.1.1 *Developer action elements:*

AVA_VAN.1.1D

The developer shall provide the TOE for testing.

7.6.1.2 *Content and presentation elements:*

AVA_VAN.1.1C

The application shall be suitable for testing.

7.6.1.3 *Evaluator action elements:*

AVA_VAN.1.1E

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

AVA_VAN.1.2E

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

AVA_VAN.1.3E

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

8 TOE Summary Specification

The following sections identify the security functions of the TOE and describe how the TSF meets each claimed SFR. They include Cryptographic Support, User Data Protection, Security Management, Privacy, Protection of the TSF, and Trusted Path/Channels. The following table defines which underlying platform of the TOE perform the capabilities described by the SFR.

8.1 Cryptographic Support

All cryptographic services for the TOE are provided by the underlying platform when the application is installed on a desktop running Windows 10 and Windows Server 2019. The specific cryptographic implementation for the Windows platform can be found in both Windows Security Target documentation referenced in section 1.1.5.

8.1.1 **FCS_CKM_EXT.1**

The TOE does not perform any asymmetric key generation. Sensor Group certificates, for HTTPS/TLS communication, are installed as part of the installation and are generated on the management server, not on the TOE's host endpoint system.

8.1.2 **FCS_RBG_EXT.1**

The TOE does not call on DRBG services. The TOE invokes the Windows platform for encrypted storage of credentials and trusted communications. Therefore, the Windows platform calls on the DRBG services required.

8.1.3 **FCS_STO_EXT.1**

The TOE invokes the platform for securely storing certificates and keys in the Windows key storage repository. The following table describes what keys were stored, their purpose, location, and origination.

Key Material	Purpose	Storage Location	Origin
Management Server Certificate (public key)	Used by the host platform OS to validate the management server during the HTTPS/TLS channel establishment.	Microsoft Key Storage Provider; located in LocalMachine\CarbonBlack certificate store.	Pre-generated by customer and installed on management server. The management server incorporates into the TOE installer. Installed on host endpoint system as part of TOE installation procedures.
Sensor Group Certificate (TLS Certificate Private Key)	Used by host platform OS to identify the TOE's Sensor Group during HTTPS/TLS channel establishment to the management server	Microsoft Key Storage Provider; located in LocalMachine\CarbonBlack certificate store.	Generated on the management server and is incorporated into the TOE installer. The certificate is installed as part of the application installation execution.

			In case of a Sensor Group change, the management server would generate a new Sensor Group Certificate. These new certificates are transferred to the TOE as part of the periodic polling cycle information exchange. The TOE application will install the new certificate and private key.

Table 12: Certificate and Key Store

The CA certificate, that is part of the server certificate chain and is installed on the management server, must be manually installed as a pre-requisite on any endpoint system that the TOE will be installed.

8.2 User Data Protection

8.2.1 FDP_DAR_EXT.1

As discussed in FCS_STO_EXT.1, The TOE invokes the platform for securely storing certificates and keys based on the Windows key storage.

There is no interactive user interface provided by the TOE and therefore there is no user data generation possible from the TOE. The TOE itself does record the collected event information into log files that is stored on the system hard drive. This information has the potential of containing information gathered from the sensitive data files such as system logs or memory dumps, as defined in FDP_DEC_EXT.1.2. Therefore, the TOE relies on the underlying operating system for encryption of this potentially sensitive data. Windows platforms do not provide data-at-rest encryption. Therefore, additional programs like BitLocker or Encrypting File System (EFS) must be used. BitLocker was enabled to provide the data-at-rest encryption for the tested configuration.

8.2.2 FDP_DEC_EXT.1

The TOE restricts its access to the endpoint system's network connectivity hardware resource for the purposed defined in FDP_NET_EXT.1.

The only endpoint system's sensitive information that the TOE requires access to are the following 2 items:

- Windows system logs (i.e., event logs)
- Memory dumps

These files are considered sensitive as they could include possible user credential information exposure or other infrastructure information that an enterprise environment may consider sensitive.

8.2.3 FDP_NET_EXT.1

The TOE requires network access to communicate with the operational environment's management server. During startup the TOE invokes the platform to establish a persistent HTTPS/TLS connection to the management server for the purpose of periodically transmitting the collected information about the endpoint system and retrieve configuration updates. This periodic polling is approximately every 30 seconds. If the HTTPS/TLS connection is interrupted, the TOE will automatically invoke the platform to re-establish the connection.

During a polling cycle, the TOE

- transmits endpoint host OS telemetry data (processes/threads being created, filesystem activity, registry activity, etc.), system logs, and/or memory dumps in accordance with active data collection configuration
- retrieves data collection configuration settings/updates, configuration changes such as whitelist/blacklist definitions, Sensor Group change, and TOE software updates.

The TOE only invokes the OS for a HTTP/TLS client connection, therefore requires no listening port (required when a platform is acting as a server) to be established.

8.3 Identification and Authentication

8.3.1 FIA_X509_EXT.1 and FIA_X509_EXT.2

The TOE platform performs certificate validation for certificates used for HTTPS/TLS communications. The following is checked in order to determine if a given certificate is valid:

- 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 OCSP as specified in RFC 6960
- 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.

The certificate validation service will ensure that all certificate paths terminate with a trusted root CA certificate and that all CA certificates include the basicConstraints extension with the CA flag set to TRUE. Certificate revocation status is validated using OCSP. The certificate validation service will also ensure that the extendedKeyUsage field is properly set for all certificates depending on their intended usage.

The management server certificate chain is manually installed on the host platform, via OS level commands, as a pre-requisite for installing the TOE. The Sensor Group certificate is installed as part of the installation of the TOE and is installed into the Windows certificate store.

The HTTPS/TLS implementation will automatically reject a certificate if a connection to the OCSP cannot be established and if the certificate is found to be invalid in any way, including if the revocation status is returned as unknown or revoked.

8.4 Security Management

8.4.1 FMT_CFG_EXT.1

The TOE does not provide an interactive user interface and therefore, does not require user credentials (default or otherwise) to operate. The TOE operates as a Windows service. In the evaluated configuration, the only credentials used are the certificates used to support HTTPS/TLS communication between the TOE and the management server. The management server certificate chain is manually installed as a pre-requisite for installing the TOE. The Sensor Group certificate is installed as part of the installation of the TOE and is installed into the Windows certificate store.

The TOE software is automatically installed with the appropriate file permissions to prevent unauthorized access to the binaries, configuration settings, and data.

8.4.2 FMT_MEC_EXT.1

All configuration settings, as established by the management server, are stored on the TOE according to the Windows platform best practices. The credential data is stored according to FCS_STO_EXT.1. Additionally, the TOE application uses the Windows Registry (HKLM/Software/CarbonBlack) to store application configuration settings.

The TOE installs the following registry subkeys:

- The subkey used for the Windows user-mode service (cb.exe) is located at HKLM\SYSTEM\CurrentControlSet\Services\CarbonBlack. This only contains parameters for loading the TOE as a service by the OS. There are no TOE configuration parameters under this registry key.
    - The subkey for the Windows user-mode service (cb.exe) application configuration is located at HKLM\SOFTWARE\CarbonBlack\config. This is the only location which stores the data collection configuration as well as configuration information such as the Sensor Group.
- The subkey used for the file system driver (cbk7.sys) is located at HKLM\SYSTEM\CurrentControlSet\Services\carbonblackk. This only contains parameters for loading the driver by the OS. There are no TOE configuration parameters under this registry key.

- The subkey used for the network filter driver (cbstream.sys) is located at HKLM\SYSTEM\CurrentControlSet\Services\cbstream. This only contains parameters for loading the driver by the OS. There are no TOE configuration parameters under this registry key.
- The subkey used for the ELAM driver (cbedrelam.sys) is located at HKLM\SYSTEM\CurrentControlSet\Services\Cbedrelam. This only contains parameters for loading the driver by the OS. There are no TOE configuration parameters under this registry key.

8.4.3 **FMT_SMF.1**

Both the local administrator and endpoint user are considered the owner or user of the endpoint device for which the TOE is installed. A typical endpoint user does not have any management functionality. However, the local administrator (Windows OS administrator) can use OS commands to verify the application version, start and stop the “CarbonBlack” service, uninstall the application, as well as load and unload the kernel drivers.

Once the application is installed on the endpoint, it is started automatically and operates immediately with no user action required. The Windows system will automatically start the TOE during the system startup of the endpoint system.

A local OS administrator can execute “c:\Windows\CarbonBlack\cb.exe -v” to verify the application version versus using a Windows command/function.

All other local management capabilities of the TOE are accomplished using the Window interfaces such as Windows Services (start/stop TOE), appwiz.cpl (uninstall TOE), or invoked by the management server.

The enterprise administrator can modify the data collection policy from the management server which controls what system data (enable/disable) is collected and ultimately transmitted to the management server. Upon receiving the configuration change, the TOE immediately enforces the changes. Additionally, a local administrator can stop/start the service using the Windows Services interface. This effectively enables/disables the transmission of any information describing the system’s hardware, software, or configuration.

8.5 Privacy

8.5.1 **FPR_ANO_EXT.1**

The TOE application does not collect personally identifiable information (PII) for administrators or users. Therefore, the TOE application does not transmit PII data over the network.

8.6 Protection of the TSF

8.6.1 **FPT_AEX_EXT.1**

The TOE is a Windows service and does not operate in user space. The TOE does not provide a user interface for users to generate user-modifiable files. It is compatible with operating with Windows Defender Exploit Guard Exploit Protection configured with the following minimum mitigations enabled;

- Control Flow Guard (CFG),

- Randomize memory allocations (Bottom-Up ASLR),
- Export address filtering (EAF),
- Import address filtering (IAF), and
- Data Execution Prevention (DEP).

The TOE is compiled with the following flags to ensure anti-exploitation capabilities are enabled for address space layout randomization (ASLR), Data Execution Prevention (DEP), and buffer overflow protection.

Flag	Description	cb.exe
/DYNAMICBASE (Use address space layout randomization)	This linker option enables the building of an executable image that can be loaded at different locations in memory at the beginning of execution. This option also makes the stack location in memory much less predictable.	Yes
/GS (Buffer Security Check)	Instructs the compiler to insert overrun detection code into functions that are at risk of being exploited. When an overrun is detected, execution is stopped. By default, this option is on.	Yes
/guard (Enable Control Flow Guard)	Causes the compiler to analyze control flow for indirect call targets at compile time, and then to insert code to verify the targets at runtime.	Yes
/HIGHENTROPYVA	Specifies whether the executable image supports high-entropy 64-bit address space layout randomization (ASLR).	Yes
/NXCOMPAT, /NXCOMPAT (Compatible with Data Execution Prevention)	These compiler and linker options enable Data Execution Prevention (DEP) compatibility. DEP guards the CPU against the execution of non-code pages.	Yes

8.6.2 FPT_API_EXT.1

When the TOE is installed on the Windows OS and the TOE uses only the following supported Windows platform APIs in order to function.

Windows DLL containing system calls used by TOE	
advapi32.dll	ntdll.dll
api-ms-win-core-version-l1-1-0.dll	psapi.dll
bcrypt.dll	rpcrt4.dll
combase.dll	secur32.dll
crypt32.dll	shell32.dll
dbgcore.dll	shlwapi.dll
dnsapi.dll	user32.dll
fltlib.dll	userenv.dll
gdi32.dll	winhttp.dll
gdiplus.dll	wintrust.dll
iphlpapi.dll	ws2_32.dll
kernel32.dll	wtsapi.dll
ncrypt.dll	

Table 13: API listing

The above DLL files are contained in the following Microsoft API categories identified at the Windows developer website: https://docs.microsoft.com/en-us/windows/

- Data access and storage API
- Devices API
- Diagnostics API
- Graphics and Multimedia API
- Installable file systems DDI reference
- Networking and Internet API
- Security and Identity API
- System Admin and Management API
- System Services API
- Windows Driver Kit
- Windows Environment (Shell) API
- Windows Software Development Kit (SDK)

8.6.3 FPT_IDV_EXT.1

The TOE is versioned using "major.minor.patch.build" methodology. Major version updates happen when incompatible API changes occur, minor version updates happen when backwards-compatible functionality is added, and patch version updates happen when backwards-compatible bug fixes are implemented. Additional labels for pre-release and build metadata are available as extensions to the major version updates.

8.6.4 FPT_LIB_EXT.1

The TOE is packaged with several third-party open source libraries in order to function.

When the TOE is installed on the Windows OS, the TOE uses only the following third-party statically linked libraries in order to function.

- Google protobuf
- zlib

The TOE does not install any third-party dynamic libraries.

8.6.5 FPT_TUD_EXT.1 and FPT_TUD_EXT.2

The TOE application is packaged in the installer.exe and msi.exe formats for the Windows OS platform. The application software is digitally signed using a VMware Carbon Black commercial CA certificate (authorized source) and then cross-signed by Microsoft's WHQL signing. The software is verified by the management server prior to being made available for installation on an endpoint system. Unless the management server has made an update package available for download, the TOE does not automatically modify, replace, or update its own binaries or executable files. The TOE supports functionality for both manual or automatic (default) updates.

If the TOE has been configured for automatic updates:

During each periodic polling cycle the TOE application checks the management server's check-in response for the upgrade request object to see if there is a new update available. That object carries the information to upgrade including the sensor package name (binary) to use.

If the management server has set the upgrade request object, the TOE will download the specified sensor package (binary) and execute the update package. The update package invokes the OS APIs to validate the certificate chain (WinVerifyTrust) and install the update correctly.

If the management server has not set the upgrade request object, then there is no further upgrade action accomplished by the TOE.

If the TOE has been configured for manual updates:

The OS administrator must navigate to the management server and authenticate as an enterprise administrator and manually select the update package for downloading to the endpoint machine. The management server will only show allowed versions available for download. The list of available update packages contains the version number as part of the displayed filename. Once the update package has been downloaded, the OS administrator can execute the update package. The update package invokes the OS APIs to validate the certificate chain (WinVerifyTrust) and install the update correctly.

After the successful installation of the update package, the TOE will report that it is operating with updated version during the next polling cycle to the management server. The management server will then reset the upgrade request object to show that there is no update available.

Uninstalling the TOE removes all traces of the application that were created as part of the installation.

The TOE's OS platform provides multiple ways for the local OS administrator to check the version of the TOE that is currently running on the machine.

- appwiz.cpl – once the programs and features windows display, the scroll through list of applications until "VMware Carbon Black EDR Sensor" is displayed and the version number will be displayed to the far right.
- Properties – by right clicking on the cb.exe and selecting properties the version will be displayed in the Details tab.
- Installation package download – version number is contained within filename.
- cb.exe – execute "c:\Windows\CarbonBlack\cb.exe -v" and the version will be displayed.

8.6.5.1 *Timely Security Updates*

As part of providing timely security updates, VMware Carbon Black provides customers with a support section on CarbonBlack.com where they have the ability to submit support issues through the User Exchange link. High severity issues can result in a patch release as soon as remediation is available. Lower severity issues will be incorporated into the next scheduled release. Security fixes will be released as new packages in the same manner as any feature updates (see discussion on FPT_TUD_EXT.1 above). The TOE installation package contains all third-party components that are required. The end customer should never attempt to update the third-party packages. Any implementation flaws are expected to be addressed within 90 days of reporting. Customers are notified of security-related fixes on the VMware Carbon Black customer portal.

8.7 Trusted Path/Channels

8.7.1 FTP_DIT_EXT.1

The TOE invokes the platform’s WinHTTP to establish the trusted channel (HTTPS session over TLS v1.2) between the TOE and the management server. These protocols are used to protect the data traversing the channel from disclosure and/or modification. The platform only acts as a HTTPS/TLS client on behalf of the TOE.