Carbon Black Endpoint Detection and Response (EDR) System

Security Target

ST Version: 1.0
June 26, 2026

Carbon Black by Broadcom
3421 Hillview Ave
Palo Alto, CA 94304

Prepared By:

Booz | Allen | Hamilton
delivering results that endure

Cyber Assurance Testing Laboratory
1100 West Street
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 4
1.1.5 Reference 4
1.2 TOE Reference 5
1.3 TOE Overview 5
1.4 TOE Type 7
2 TOE Description 9
2.1 Evaluated Components of the TOE 9
2.2 Components and Applications in the Operational Environment 9
2.3 Excluded from the TOE 10
2.3.1 Not Installed 10
2.3.2 Installed but Requires a Separate License 10
2.3.3 Installed Functionality Excluded from the Evaluation 10
2.4 Physical Boundary 10
2.4.1 Hardware 10
2.4.2 Software 10
2.5 Logical Boundary 10
2.5.1 Security Audit 11
2.5.2 Cryptographic Support 11
2.5.3 User Data Protection 11
2.5.4 Host Agent 11
2.5.5 Identification and Authentication 11
2.5.6 Security Management 12
2.5.7 Privacy 12
2.5.8 Protection of the TSF 12
2.5.9 Trusted Path/Channel 12

3 Conformance Claims 13
3.1 CC Version 13
3.2 CC Part 2 Conformance Claims 13
3.3 CC Part 3 Conformance Claims 13
3.4 PP Claims 13
3.5 Package Claims 14
3.6 Package Name Conformant or Package Name Augmented 14
3.7 PP-Configuration Conformance Claims 14
3.8 Conformance Claim Rationale 14
3.9 Technical Decisions 15
4 Security Problem Definition 18
4.1 Threats 18
4.2 Organizational Security Policies 19
4.3 Assumptions 19
4.4 Security Objectives 19
4.4.1 TOE Security Objectives 19
4.4.2 Security Objectives for the Operational Environment 21
4.5 Security Problem Definition Rationale 21
5 Extended Components Definition 22
5.1 Extended Security Functional Requirements 22
5.2 Extended Security Assurance Requirements 22
6 Security Functional Requirements 23
6.1 Conventions 23
6.2 Security Functional Requirements Summary 23
6.3 Security Functional Requirements 25
6.3.1 Class FAU: Security Audit 25
6.3.2 Class FCS: Cryptographic Support 27
6.3.3 Class FDP: User Data Protection 29
6.3.4 Class FHA: Host Agent 31
6.3.5 Class FIA: Identification and Authentication 32
6.3.6 Class FMT: Security Management 33

6.3.7 Class FPR: Privacy 36
6.3.8 Class FPT: Protection of the TSF 36
6.3.9 Class FTP: Trusted Path/Channel 39
6.4 Statement of Security Functional Requirements Consistency 39
7 Security Assurance Requirements 39
7.1 Class ASE: Security Target evaluation 40
7.1.1 ST introduction (ASE_INT.1) 40
7.1.2 Conformance claims (ASE_CCL.1) 41
7.1.3 Security objectives for the operational environment (ASE_OBJ.1) 42
7.1.4 Extended components definition (ASE_ECD.1) 43
7.1.5 Stated security requirements (ASE_REQ.1) 44
7.1.6 TOE summary specification (ASE_TSS.1) 45
7.2 Class ADV: Development 45
7.2.1 Basic Functional Specification (ADV_FSP.1) 45
7.3 Class AGD: Guidance Documentation 46
7.3.1 Operational User Guidance (AGD_OPE.1) 46
7.3.2 Preparative Procedures (AGD_PRE.1) 47
7.4 Class ALC: Life Cycle Support 48
7.4.1 Labeling of the TOE (ALC_CMC.1) 48
7.4.2 TOE CM Coverage (ALC_CMS.1) 48
7.4.3 Timely Security Updates (ALC_TSU_EXT.1) 49
7.5 Class ATE: Tests 49
7.5.1 Independent Testing - Conformance (ATE_IND.1) 49
7.6 Class AVA: Vulnerability Assessment 50
7.6.1 Vulnerability Survey (AVA_VAN.1) 50
8 TOE Summary Specification 51
8.1 Security Audit 51
8.1.1 [EDRMOD] FAU_ALT_EXT.1 51
8.1.2 [EDRMOD] FAU_COL_EXT.1 51
8.1.3 [EDRMOD] FAU_GEN.1/EDR 51
8.1.4 [HAMOD] FAU_GEN.1/HA 52

8.1.5 [HAMOD] FAU_STO_EXT.1 52
8.2 Cryptographic Support 52
8.2.1 [App PP] FCS_CKM_EXT.1(Server), FCS_CKM_EXT.1(Sensor), and FCS_CKM.1/AK 52
8.2.2 [App PP] FCS_CKM.2 53
8.2.3 [App PP] FCS_RBG_EXT.1(Server) and FCS_RBG_EXT.1(Sensor) 53
8.2.4 [App PP] FCS_STO_EXT.1(Server) and FCS_STO_EXT.1(Sensor) 53
8.3 User Data Protection 54
8.3.1 [App PP] FDP_DAR_EXT.1 54
8.3.2 [App PP] FDP_DEC_EXT.1(Server) and FDP_DEC_EXT.1(Sensor) 55
8.3.3 [App PP] FDP_NET_EXT.1(Server) and FDP_NET_EXT.1(Sensor) 55
8.3.4 [HAMOD] FDP_NET_EXT.2 56
8.4 Host Agent 56
8.4.1 [HAMOD] FHA_CHA_EXT.1 56
8.4.2 [HAMOD] FHA_COL_EXT.1 57
8.4.3 [HAMOD] FHA_HAD_EXT.1 57
8.5 Identification and Authentication 57
8.5.1 [EDRMOD] FIA_AUT_EXT.1 57
8.5.2 [EDRMOD] FIA_PWD_EXT.1 57
8.5.3 [App PP] FIA_X509_EXT.1 and FIA_X509_EXT.2 57
8.6 Security Management 58
8.6.1 [App PP] FMT_CFG_EXT.1 58
8.6.2 [App PP] FMT_MEC_EXT.1 59
8.6.3 [App PP] FMT_SMF.1(Server) and FMT_SMF.1(Sensor) 59
8.6.4 [EDRMOD] FMT_SMF.1/ENDPOINT 60
8.6.5 [HAMOD] FMT_SMF.1/HA 61
8.6.6 [EDRMOD] FMT_SMF.1/HOST 61
8.6.7 [EDRMOD] FMT_SMR.1 62
8.6.8 [EDRMOD] FMT_SRF_EXT.1 62
8.6.9 [HAMOD] FMT_UNR_EXT.1 62
8.7 Privacy 63
8.7.1 [App PP] FPR_ANO_EXT.1 63

8.8 Protection of the TSF 63
8.8.1 [App PP] FPT_AEX_EXT.1 63
8.8.2 [App PP] FPT_API_EXT.1 64
8.8.3 [App PP] FPT_IDV_EXT.1 67
8.8.4 [EDRMOD] FPT_ITT.1 67
8.8.5 [App PP] FPT_LIB_EXT.1 67
8.8.6 [App PP] FPT_TUD_EXT.1(Server), FPT_TUD_EXT.1(Sensor), and FPT_TUD_EXT.2 74
8.9 Trusted Path/Channel 76
8.9.1 [App PP] FTP_DIT_EXT.1 76
8.9.2 [EDRMOD] FTP_TRP.1 76
9 Appendix A: EDR Server Third Party Libraries Obtained From SBOM 78
10 Appendix B: EDR Server User Interface Third Party Libraries Obtained From SBOM 84
11 Appendix C: EDR Sensor Third Party Libraries Obtained From SBOM 97

Table of Figures

Figure 1: TOE Boundary 6

Table of Tables

Table 1: Customer Specific Terminology 2
Table 2: CC Specific Terminology 2
Table 3: Acronym Definition 4
Table 4: Evaluated Components of the TOE 9
Table 5: Components of the Operational Environment 9
Table 6: Technical Decisions 15
Table 7: TOE Threats 18
Table 8: TOE Assumptions 19
Table 9: TOE Objectives 20
Table 10: TOE Operational Environment Objectives 21
Table 11: Security Functional Requirements for the Server 23
Table 12: Security Functional Requirements for the Sensor 24
Table 13: FMT_SMF.1.1/ENDPOINT Management Functions 34

Table 14: FMT_SMF.1.1/HA Management Functions 35
Table 15: FMT_SMF.1.1/HOST Management Functions 35
Table 16: FMT_SRF_EXT.1.1 Management Functions 36
Table 17: Certificate and Key Store 53
Table 18: FMT_SMF.1/ENDPOINT Management Functions 60
Table 19: FMT_SMF.1/ENDPOINT Management Functions 61
Table 20: FMT_SMF.1/HOST Management Functions 61
Table 21: Compiler Flags used by the TOE 63
Table 22: EDR Server APIs 64
Table 23: EDR Windows Sensor DLL files 65
Table 24: EDR Server Dynamic Libraries 67

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: Carbon Black Endpoint Detection and Response (EDR) System Security Target
ST Version: 1.0
ST Publication Date: June 26, 2026
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 product-specific terminology used throughout this ST is defined in Table 1. Technology terms that are related to the security functionality claimed by the TOE are defined in the introductory materials of the claimed Protection Profile. These tables are to be used by the reader as a quick reference guide for terminology definitions.

Table 1: Customer Specific Terminology

Term	Definition
Administrator	The administrator who has system permissions to access sensitive data and perform management functionality on the Carbon Black (CB) EDR Server.
Administrative Workstation	Any general-purpose computer, in the operational environment, that is used by an Administrator to manage the TOE remotely via a web browser.
Application Management	Management of the Sensors by an Administrator using Carbon Black EDR Server software installed on the Management Server..
Endpoint System	An operational environment device or set of devices, such as a laptop or desktop, with a Windows operating system, that has the host sensor software application installed. The physical device(s) is not part of the TOE boundary but does provide all of the cryptographic and HTTPS/TLS networking functionality.
Endpoint User	An individual who has access to the Endpoint System but is not able to manage the device.
Host Sensor	Generic term for Carbon Black EDR Windows Sensor software application. This term is synonymous with Host Agent.
Local Administrator	The Windows OS administrator who has system permissions to access sensitive data and perform management functionality on the Endpoint System.
Management Server	An operational environment server with Red Hat Enterprise Linux operating system that has the TOE's Carbon Black EDR Server software application installed. The Management Server is used for the remote management of Carbon Black EDR Windows Sensors (TOE) by an Administrator using the EDR Server web interface. The physical device is not part of the TOE boundary but does provide all of the cryptographic and HTTPS/TLS networking functionality.
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.
SOC Analyst	A user role designed to monitor and respond to suspicious or malicious activity on endpoints in Sensor Groups. SOC Analysts do not have access to data or functions for managing the server itself, such as managing users, viewing and changing server settings (including sharing settings), and viewing the server dashboard.

Table 2: CC 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.
Endpoint	A computing device that runs a general purpose OS, a mobile device OS, or network device OS. Endpoints can include desktops, servers, and mobile devices.
Endpoint Detection and Response (EDR)	Software that collects and analyzes data from endpoints for detecting, investigating, and remediating unauthorized activities on endpoints.
Host Agent	Complementary software that executes on endpoints to collect data about the endpoint and executes commands sent to the endpoint from an Enterprise Security Management (ESM) server or service. An example command sent to an endpoint could be to enforce a policy from an ESM, to collect some files, or to run an OS command.
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 of 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 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.

1.1.4 Acronyms

The acronyms used throughout this ST are defined in Table 2. This table is to be used by the reader as a quick reference guide for acronym definitions.

Table 3: Acronym Definition

Acronym	Definition
AA	Assurance Activity
API	Application Programming Interface
ASLR	Address Space Layout Randomization
CB	Carbon Black
CC	Common Criteria
DEP	Data Execution Prevention
DRBG	Deterministic Random Bit Generator
EDR	Endpoint Detection and Response
ESM	Enterprise Security Management
GUI	Graphical User Interface
HA	Host Agent
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
IT	Information Technology
LUKS	Linux Unified Key Setup
OS	Operating System
OSP	Organizational Security Policy
PII	Personally Identifiable Information
PP	Protection Profile
NIAP	National Information Assurance Partnership
RBG	Random Bit Generator
RHEL	Red Hat Enterprise Linux
SAR	Security Assurance Requirement
ST	Security Target
TD	Technical Decision
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function

1.1.5 Reference

[1] Protection Profile for Application Software, version 1.4 [App PP]

[2] PP-Module for Endpoint Detection and Response (EDR), version 1.0 [EDRMOD]

[3] PP-Module for Host Agent, version 1.0 [HAMOD]

[4] 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

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

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

[7] Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-004
[8] Red Hat Enterprise Linux 8.6 Security Target, Version 1.1, January 2024
[9] Microsoft Windows 11 Security Target, Version 0.07, October 2022
[10] Carbon Black EDR 7.9.0 User Guide, 16 December 2025
[11] Carbon Black Endpoint Detection and Response (EDR) System, Supplemental Administrative Guidance for EDR Server v7.9, v1.0, 31 December 2025
[12] Carbon Black Endpoint Detection and Response (EDR) System, Supplemental Administrative Guidance for EDR Host Sensor v7.5, v1.0, 31 December 2025

1.2 TOE Reference

The TOE is Carbon Black Endpoint Detection and Response (EDR) System consisting of the EDR Server v7.9.0 and EDR Windows Sensor v7.5.0. The Carbon Black EDR System consists of both the EDR Server residing on a Red Hat Enterprise Linux (RHEL) 8.6 OS and the EDR Windows Sensor residing on a Microsoft Windows 11 OS.

1.3 TOE Overview

The TOE is the Carbon Black Endpoint Detection and Response System, referred to as the EDR System or the TOE from this point forward. The EDR System's primary functionality is to continuously record and store endpoint activity data to aid in incident response and threat hunting activities. The minimum configuration for this evaluation is one Carbon Black Endpoint Detection and Response Server (EDR Server) and one Carbon Black Endpoint Detection and Response Windows Sensor (Host Sensor) installed on an endpoint. Additional Endpoint Systems with a Host Sensor application installed as part of the operational configuration do not affect the validity of the functional claims made within this document and the Common Criteria certification.

The EDR Server is a software application that is installed on a RHEL 8.6 system with Linux Unified Key Setup (LUKS) encrypted partitioning enabled. As part of the EDR System, the EDR Server receives Endpoint System event data from a Host Sensor for indexing, analyzing, and storing the event data. The EDR Server allows an Administrator to create and deploy Sensor Groups, configure the Host Sensor, configure the data collection policy that the Host Sensor will enforce, update the Host Sensor software, and uninstall a Host Sensor. The EDR Server and Host Sensor communicate during each Host Sensor's periodic check-ins and when a possible threat is detected.

The Host Sensor is a host agent software application that operates on top of Microsoft Windows 11 with BitLocker (or equivalent) enabled. The Host Sensor can be installed on administratively defined network endpoints, such as laptops, desktops, and servers. As part of the EDR System, the Host Sensor operates as a Windows service to perform its function of observing and reporting on system-level behavior. The Host Sensor stores the collected event data into its own data storage on the local file system. The event data is then transferred from the Host Sensor, using a HTTPS session over TLS v1.2 (HTTPS/TLS) trusted channel, to the EDR Server. The data collection policy configuration settings are stored in the Windows registry and changes to the data collection policy can only be initiated by the EDR Server.

The TOE is administered through a web user interface (Web GUI) via a Google Chrome web browser. Through the Web GUI, an Administrator has the ability to access the EDR Server to configure and

manage the TOE. In the evaluated configuration, as depicted in Figure 1, the TOE's evaluation scope is the EDR Server software and the Host Sensor software. The illustration in Figure 1 is meant to serve as an example of one kind of deployment configuration with a single Host Sensor. A deployment configuration with more than one Host Sensor is acceptable.

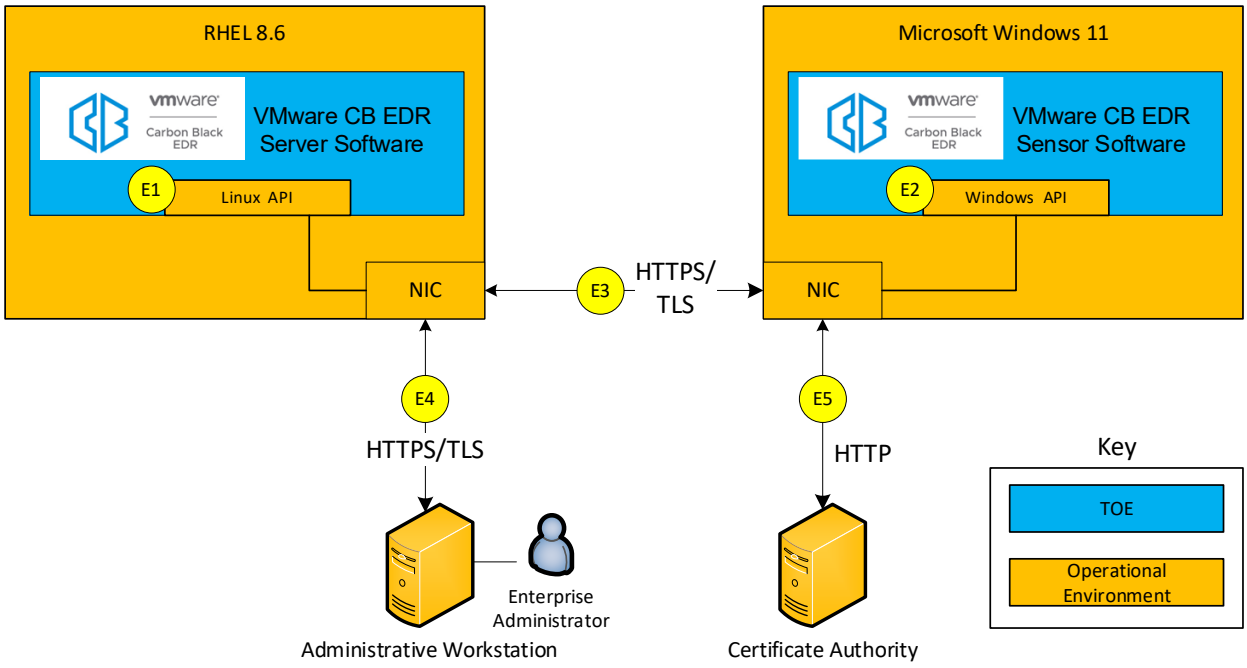


Figure 1: TOE Boundary

Supporting Environment Interfaces to the TOE depicted in Figure 1:

- **E1: EDR Server application to OS** –The TOE leverages operating system callbacks for file system access, storage and indexing of received Host Sensor event data, and accessing network drivers. The TOE invokes the OS for network access and cryptographic support.
- **E2: Host Sensor application to OS** – 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.
- **E3: Host Sensor platform to EDR Server platform** – The EDR Server platform responds to a Host Sensor platform's HTTPS/TLS connection request in order to establish a trusted communication channel. This channel is used for receiving the collected event data from the Host Sensor and provides configuration changes and updates for the Host Sensor to pull from the EDR Server. The Host Sensor platform (OS) is the HTTPS/TLS client and performs X.509 certificate validation in support of the non-mutually authenticated HTTPS session over TLS v1.2 communications. The EDR Server platform is a HTTPS/TLS server for this interface.
- **E4: Web Browser to EDR Server platform** – A remote Administrator uses a web browser from an administrative workstation to access the TOE's Web GUI. This transmission is secured using HTTPS session over TLS v1.2 (HTTPS/TLS). It provides a graphical interface to the TOE in order to perform remote administrative activities or to view reports or other graphical displays of

system data that is collected by a Host Sensor. Users authenticate to the Web GUI using a username and password. The EDR Server platform is a HTTPS/TLS server for this interface.

- **E5: Host Sensor to Certificate Authority** – The EDR Windows Sensor 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 an Endpoint Detection and Response software application where the EDR Server software application is deployed on a RHEL OS and the Host Sensor software application is deployed on a Windows OS. An EDR software product has ESM system functionality requirements defined in the EDR and Host Agent module protection profiles. Therefore, an EDR product is also characterized as an ESM application.

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 provides administration to an enterprise-level system visibility for the IT infrastructure. The TOE is installed on a platform OS and communicates with an Endpoint System that collects event data.

The [EDRMOD] states: "The TOE boundary for the EDR encompasses all the software from the TOE vendor that represents the server or enterprise management side of the EDR system. This will typically, but not always, be software running behind a web application or dashboard, and possibly with other software services running to send and receive data with a Host Agent. The EDR may also make use of a database to store collected and analyzed data. Any database software itself is outside the scope of the TOE, as is any web server software used to serve a web application or dashboard, and the underlying operating system or cloud platform."

The [HAMOD] states: "The boundary for the Host Agent includes all processes, all modules, and libraries bundled with the Host Agent. The Host Agent can run as a daemon or service on the platform but is not required to. The Host Agent is not expected to have a local or remote Graphical User Interface (GUI) for administration but having such an interface is not precluded by this PP-Module. It is expected that Host Agents will be managed by their associated ESM server or the underlying platform. The TOE boundary includes the communications channel with other Host Agents, an ESM server, or a cloud service. The platform operating system or execution environment upon which the Host Agent is executing is outside the scope of a Host Agent evaluation. The figures below show some sample Host Agents but are not inclusive of every possible Host Agent design."

The EDR and Host Agent TOE types are justified because the TOE is an ESM application that relies on the use of Host Agents running on Endpoint Systems to monitor and remediate potentially malicious

activity.

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:

Table 4: Evaluated Components of the TOE

Component	Definition
Carbon Black Endpoint Detection and Response Server 7.9.0 (EDR Server)	An application that is installed on a Red Hat Enterprise Linux platform. This application has a Web GUI to allow TOE users to manage the operations of the EDR System as well as review collected data for threats. The application communicates with one or more Host Sensors to receive collected data as well as send configuration information or remediation actions for the Host Sensor to process.
Carbon Black Endpoint Detection and Response Windows Sensor 7.5.0 (Host Sensor)	An application that is installed on a Windows platform which collects event data from its host platform, reports the data back to the TOE, and performs remediation actions based upon commands from the EDR Server.

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:

Table 5: Components of the Operational Environment

Component	Definition
Red Hat Enterprise Linux (RHEL) 8.6 Server	The host platform with the RHEL operating system environment that the EDR Server application is installed on.**
Endpoint System with Microsoft Windows 11 Pro edition (22H2)	The host platform along with the operating system installed that the Host Sensor application is installed on.**
Administrative Workstation	Any general-purpose computer that is used by an Administrator to manage the TOE 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 Endpoint System with Host Sensor to validate the server X.509v3 certificate from the EDR Server's platform.

**NOTE: It is expected that the TOE is operating on a Common Criteria certified operating system and platform based on the Red Hat Enterprise Linux (RHEL) 8.6 evaluation (VID11309) and Microsoft Windows 11, Windows Server 2022, Windows 11 Pro and other Windows OSes evaluations (ES scheme).

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 added 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 optional components that are omitted from the installation process.

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 functionalities that are satisfied by the Security Functional Requirements in the claimed Protection Profile. Therefore, the following are considered out of scope of the evaluated configuration because there are no SFRs in the PP that allow this functionality to be claimed:

- The EDR System includes a number of capabilities in support of its primary function, such as the collection, storage, and analysis of host platform data, that are outside the scope of the claimed Protection Profile.
- The EDR System supports having multiple EDR servers in a cluster/minion configuration where a host sensor can communicate with multiple EDR servers. This cluster/minion configuration, along with the communications between EDR servers is not part of the evaluated configuration. Each host sensor is only communicating with one EDR server and host sensors do not talk to other host sensors.

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 EDR Server version 7.9.0 (running on Red Hat Enterprise Linux (RHEL) 8.6) and Host Sensor version 7.5.0 (running on Microsoft Windows 11) as well as both component's respective configuration data.

2.5 Logical Boundary

The TOE is comprised of several security features. Each of the security features identified above consists of several security functionalities, as identified below.

1. Security Audit
2. Cryptographic Support
3. User Data Protection

4. Host Agent
5. Identification and Authentication
6. Security Management
7. Privacy
8. Protection of the TSF
9. Trusted Path/Channel

2.5.1 Security Audit

The EDR Server generates and displays alerts to authorized users when potentially unauthorized activity is detected on enrolled endpoints. The EDR Server collects data from the endpoint that the Host Sensor is installed on. Both the EDR Server and the Host Sensor contain mechanisms to generate audit data for a set of predefined events. The Host Sensor relies on Windows Event Logger to store audit events.

2.5.2 Cryptographic Support

The EDR Server invokes the underlying platform OS to perform all cryptographic services including establishing HTTPS sessions over TLSv1.2 trusted communications and hashing user password credentials for storage.

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

2.5.3 User Data Protection

The TOE application restricts its access to the EDR Server platform's and Host Sensor platform's network connectivity resources. The TOE also restricts the Host Sensor's sensitive data access to system logs and memory dumps stored on the Endpoint System. Network activity is restricted to establishing HTTPS/TLS connections for remote management via Web GUI to the EDR Server, and between the Host Sensor and EDR Server for reachability events. During the Host Sensor check-in reachability event, the EDR Server receives collected Endpoint System data from the Host Sensor as well as provides any configuration and software updates for the Host Sensors to pull. The EDR Server requires LUKS encrypted partitioning to protect local sensitive data storage.

2.5.4 Host Agent

Data collected on the Host Sensor is stored locally until it is forwarded to the EDR Server over a secure HTTPS/TLS channel during Host Sensor check-in reachability event. The Host Sensor interacts only with the EDR Server and no other ESM software. There is no sensor-to-sensor communication for deployments of the TOE that contain more than one Host Sensor.

2.5.5 Identification and Authentication

The EDR Server provides username/password based authentication to support administrative access to the TOE's Web GUI. Authentication passwords defined by the Administrator are only accepted by the TOE if the password meets the minimum password requirements. The Host Sensor enforces the use of X.509v3 certificates to support the establishment of all HTTPS/TLS connections to the EDR Server.

2.5.6 Security Management

The TOE provides Administrators with the ability to manage the EDR Server and Host Sensor (through the EDR Server) through a Web GUI. Each component of the TOE uses its underlying platform's recommended methods for storing and setting configuration options. During installation, the Host Sensor is automatically configured to protect itself and its data from unauthorized access.

2.5.7 Privacy

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

2.5.8 Protection of the TSF

Each TOE component is packaged as separate software that is installed on the platform and can be uninstalled/removed if needed. All updates are downloaded and installed by the Administrator using the OS software package manager. All updates for the Host Sensor are obtained through the EDR Server. The digital signature of the update is verified by the EDR Server platform during installation. The TOE does not download, replace, or modify its own binary code. The Administrator can verify the software version of the TOE from the Web GUI. The EDR Server invokes the OS platform to operate as the server in the HTTPS/TLS v1.2 communication channel with the Host Sensor.

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

2.5.9 Trusted Path/Channel

The EDR Server invokes the OS platform to operate as the server in the HTTPS/TLS v1.2 communication channel with the web browser on the administrative workstation.

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 June 26, 2026.

3.3 CC Part 3 Conformance Claims

This ST and Target of Evaluation (TOE) is Part 3 extended to include all applicable NIAP and International interpretations through June 26, 2026.

3.4 PP Claims

This ST claims exact conformance to the following Protection Profile:

- Protection Profile for Application Software, version 1.4 [App PP], 2021-10-07
- Protection Profile Module for Endpoint Detection and Response (EDR), version 1.0 [EDRMOD], 2020-10-23
- Protection Profile Module for Host Agent, version 1.0 [HAMOD], 2020-10-23

The TOE claims following Selection-Based SFRs that are defined in the appendices of the claimed PP and PP-Modules:

- FCS_CKM.1/AK
- FCS_CKM.2
- FHA_CHA_EXT.1
- FHA_COL_EXT.1
- FIA_X509_EXT.1
- FIA_X509_EXT.2
- FPT_TUD_EXT.2

The TOE claims following Optional SFRs that are defined in the appendices of the claimed PP and PP-Modules:

- none

The TOE claims following Objective SFRs that are defined in the appendices of the claimed PP and PP-Modules:

- none

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

3.5 Package Claims

This ST does not claim conformance to any Functional Packages.

3.6 Package Name Conformant or Package Name Augmented

This ST does not claim conformance to any Functional Packages.

3.7 PP-Configuration Conformance Claims

This PP-Configuration is identified as follows:

- PP-Configuration for Application Software, Endpoint Detection and Response, and Host Agent, Version 1.1, 07 April 2022
- As a shorthand reference, it can be identified as [CFG_APP-EDR-HA_V1.1]

This PP-Configuration includes the following components:

- Base-PP: Protection Profile for Application Software, Version 1.4 (PP_APP_V1.4)
- PP-Module: PP-Module for Endpoint Detection and Response, Version 1.0 (MOD_EDR_V1.0)
- PP-Module: PP-Module for Host Agent, Version 1.0 (MOD_HA_V1.0)

Conformance Statement:

To be conformant to this PP-Configuration, an ST must demonstrate Exact Conformance, as defined by the CC and CEM addenda for Exact Conformance, Selection-Based SFRs, and Optional SFRs (dated May 2017).

CC Conformance Claims:

This PP-Configuration, and its components specified in Section 1.3, are conformant to Parts 2 (extended) and 3 (extended) of Common Criteria Version 3.1, Release 5 [CC].

SAR Statement:

The SARs specified for this PP-Configuration are taken from, and identical to, those specified in the MDM Base-PP (PP_APP_V1.4).

3.8 Conformance Claim Rationale

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 EDRMOD states the following: "An EDR is enterprise management software that collects endpoint host data to detect potentially unauthorized activity on endpoints and to enable threat hunting and other incident response actions to remediate malicious behaviors. These requirements cover basic security characteristics and behaviors for EDR products; the platform on which the EDR runs may be a physical or virtual Operating System (OS), and on-premises or in a cloud environment."

The HAMOD states the following: "The requirements for the EDR are not covered in this PP-Module, however it is expected that an ESM system will evaluate against a PP-Configuration that includes both the [EDR] PP-Module and the Host Agent PP-Module. The EDR PP-Module covers the security functionality needed on the server or cloud service, and the paired Host Agent PP-Module covers the security functionality needed on the endpoint device (desktop, mobile device, etc.)."

The TOE is an EDR application which consists of both an EDR Server and at least one Host Sensor. The EDR Server runs on a RHEL 8.6 OS platform and the Host Sensor runs on a Microsoft Windows 11 OS platform. The TOE is therefore considered to be relevant to the App PP, EDRMOD, and HAMOD.

3.9 Technical Decisions

Technical Decisions that affected the SFR wording have been annotated with a Footnote. The following is a complete list of Technical Decisions that apply to the App PP, EDRMOD, and HAMOD evaluation activities that must be performed during the evaluation of this TOE:

Table 6: Technical Decisions

TD #	Title	References	Changes			Analysis to this evaluation	
			SFR	AA	Notes	NA	Reason
TD0628	Addition of Container Image to Package Format	[App PP] FPT_TUD_EXT.2.1	X	X			Updated SFR wording and wording for Android, Microsoft Windows, Apple IOS, Linux, Oracle Solaris, and Apple mac OS Tests Footnote 18
TD0650	Conformance Claim sections updated to allow for MOD_VPNC_V2.3 and 2.4	[App PP] Section 2			X		Update to section 2 to allow for PP-Module for VPN Clients, Version 2.4 to be claimed
TD0664	Testing activity for FPT_TUD_EXT.2.2	[App PP] FPT_TUD_EXT.2.2		X			Updates wording of Android and Apple OS Tests

							Insertion of "All other Platforms" Test Deletion of Microsoft Windows, Linux, Oracle Solaris, Apple MacOS Specific Tests
TD0717	Format changes for PP_APP_V1.4	[App PP] FCS_CKM.1, FCS_CKM.2, FCS_CKM.1/AK, FCS_CKM.1/PBKDF, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, FCS_COP.1/Sig, FCS_COP.1/SKC	X				Various insertions and deletions within the SFRs referenced Footnote 4, 5, 6, and 7
TD0719	ECD for PP APP V1.3 and 1.4	[App PP]			X		Extended Component Definitions appendix changes to PP_APP_v1.3 to PP_APP_v1.4.
TD0733	FMT_UNR_EXT.1 test functionality	[HAMOD] FMT_UNR_EXT.1, MOD_HA_V1.0-SD		X			Updates to test activity
TD0734	Clarification on optional functions within FMT_SRF_EXT.1	[EDRMOD] FMT_SRF_EXT.1, MOD_EDR_V1.0-SD		X	X		Updates to application note and test 4
TD0735	Guidance evaluation activity clarification for FAU_ALT_EXT.1.3	[EDRMOD] FAU_ALT_EXT.1.3, MOD_EDR_V1.0-SD		X			Updates to guidance evaluation activity
TD0736	Number of elements for iterations of FCS_HTTPS_EXT.1	[App PP] FCS_HTTPS_EXT.1.3/ Server	X	X		X	Updates to SFR and evaluation activities Not Claiming HTTPS
TD0743	FTP_DIT_EXT.1.1 Selection exclusivity	[App PP]	X		X		Updates to SFR wording and application note Footnote 19
TD0744	Corrections to Test Issues with FAU_COL_EXT.1.1	[EDRMOD] FAU_COL_EXT.1.1, MOD_EDR_V1.0-SD	X	X	X		Updates to SFR wording, application note and test 4a and 4b and 5 Footnote 2
TD0745	Corrections to FIA_AUT_EXT.1 Tests	[EDRMOD] FAU_AUT_EXT.1, MOD_EDR_V1.0-SD		X			Updates to test 1 and test2 wording.
TD0747	Configuration Storage	[App PP] FMT_MEC_EXT.1		X			Updates to Android test

	Option for Android						activity
TD0749	Update to FHA_CHA_EXT.1	[HAMOD] FHA_CHA_EXT.1	X	X			Updates to SFR wording, application note and test activities Footnote 10, 11
TD0756	Update for platform-provided full disk encryption	[App PP] FDP_DAR_EXT.1		X			Updates to test activity
TD0757	Clarifications for changes in Host Agent enrollment status in EDR PP-Module	[EDRMOD] FAU_ALT_EXT.1.1, FAU_GEN.1/EDR, MOD_EDR_V1.0-SD	X	X			Updates to SFR and evaluation activities Footnote 1, 3
TD0776	Clarifications to FMT_SMF.1/ENDPOINT	[EDRMOD] FMT_SMF.1/ENDPOINT, MOD_EDR_V1.0-SD	X	X	X		Updates to SFR wording, application note and test activities Footnote 12, 13, 14
TD0778	Clarification of FMT_SMF.1/HOST SFR	[EDRMOD] FMT_SMF.1/HOST, MOD_EDR_V1.0-SD	X	X	X		Updates to SFR wording, application note and test activities Footnote 15, 16, and 17
TD0780	FIA_X509_EXT.1 Test 4 Clarification	[App PP] FIA_X509_EXT.1.1		X			Updates to Test 4 and associated Note
TD0798	Static Memory Mapping Exceptions	[App PP] FPT_AEX_EXT.1.1		X			Updates to TSS and testing evaluation activities
TD0815	Addition of Conditional TSS Activity for FPT_AEX_EXT.1.5	[App PP] FPT_AEX_EXT.1.5		X			Updates to TSS and testing evaluation activities
TD0822	Correction to Windows Manifest File for FDP_DEC_EXT.1	[App PP] FDP_DEC_EXT.1.1, FDP_DEC_EXT.1.2		X			Updates to testing evaluation activities
TD0823	Update to Microsoft Windows Exploit Protection link in FPT_AEX_EXT.1.3	[App PP] FPT_AEX_EXT.1.3		X			Updates to testing evaluation activities
TD0844	Addition of Assurance Package for Flaw Remediation V1.0 Conformance Claim	[App PP] Conformance Claims			X		Update to Section 2 Conformance Claims
TD0859	Clarification for TSS Activity for FIA_PWD_EXT.1	[EDRMOD] FIA_PWD_EXT.1		X			Updates to TSS evaluation activities.
TD0865	Consistency of Cryptographic Key Sizes	[App PP] FCS_STO_EXT.1.1, FCS_CKM.1.1/PBKDF	X			X	Updates to SFR Change does not impact

		, FCS_COP.1.1/SKC, FCS_CKM.1.1/SK					current evaluation claims Footnote 8 and 9
TD0914	Addition of PKG_TLS_V2.0 to Conformance Claims	[App PP] PP_APP_v1.4			X	X	PP is not claiming conformance to the TLS package.
TD0931	Clarification when CTR_DRBG is Selected for FCS_RBG_EXT.2.2 in PP_APP_V1.4	[App PP] FCS_RBG_EXT.2.2	X			X	Not claiming FCS_RBG_EXT.2
TD0945	Adding FIPS 186-5 in PP_APP_V1.4	[App PP] PP_APP_v1.4	X				Footnote 6
TD0947	Adding FIPS 186-5 in MOD_EDR_V1.0	[EDRMOD] FMT_TRM_EXT.1.1				X	Not claiming FMT_TRM_EXT.1
TD0948	Adding FIPS 186-5 in MOD_HA_V1.0	[HAMOD] FMT_POL_EXT.1.1				X	Not claiming FMT_POL_EXT.1
TD0964	Clarification to FMT_MEC_EXT.1 Windows Test	[App PP] FMT_MEC_EXT.1		X			Testing updated or windows based platform
TD1031	Addition of PP_APP_V1.4 to Base-PPs	[EDRMOD] [HAMOD] [Reference: Section 1.1			X		list of Base-PPs in Section 1.1 Overview for MOD_EDR_V1.0 and MOD_HA_V1.0:

4 Security Problem Definition

4.1 Threats

This section identifies the threats against the TOE. These threats have been taken from the App PP, PP-Module for EDR, and PP-Module for Host Agent.

Table 7: TOE Threats

Threat	Threat Definition
[EDRMOD] T.CREDENTIAL_REUSE	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may guess or harvest legitimate credentials from the EDR, endpoints, or insecure network activity.
[HAMOD] T.DATA_LOSS	A Host Agent can be susceptible to data loss during periods when connectivity to the ESM system is not present.
[App PP] 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.
[EDRMOD]	An attacker is a legitimate privileged user with access to change the

T.MISCONFIGURATION	configuration of the EDR's security capabilities or is not a legitimate privileged user trying to access without proper authorization. Attackers may attempt to hide malicious activities from other privileged users.
[App PP] 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.
[App PP] 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.
[App PP] T.PHYSICAL_ACCESS	An attacker may try to access sensitive data at rest.
[HAMOD] T.TAMPER	A Host Agent can be susceptible to tampering by unprivileged users who may try to uninstall or disrupt the Host Agent's ability to function properly.

4.2 Organizational Security Policies

There are no Organizational Security Policies in the App PP, PP-Module for EDR, and PP-Module for Host Agent.

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, PP-Module for EDR, and PP-Module for Host Agent.

Table 8: TOE Assumptions

Assumption	Assumption Definition
[EDRMOD] A.CONNECTIVITY	The EDR relies on network connectivity to carry out its management activities. The OE will provide reliable network connectivity for the EDR to operate. The EDR will robustly handle occasional instances when connectivity is unavailable or unreliable.
[App PP] 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.
[App PP] 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.
[App PP] 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.

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 as defined by the App PP, PP-Module for EDR, and PP-Module for Host Agent.

Table 9: TOE Objectives

Objective	Objective Definition
[EDRMOD] O.ACCOUNTABILITY	The TOE must provide logging facilities which record management actions undertaken by identified and authenticated management users.
[HAMOD] O.ACCOUNTABILITY	Conformant Host Agents ensure that information exists that allows administrators to discover unintentional issues with the configuration and operation of the Host Agent.
[HAMOD] O.DATA_RECORDER	Conformant Host Agents will collect security-relevant data from a target entity in the Operational Environment and transmit it to a trusted entity for further processing. They will also implement mechanisms to ensure that collected data is retained in the event that the trusted channel is unavailable to prevent data loss.
[EDRMOD] O.EDR_MANAGEMENT	The TOE must facilitate authorized management by the enterprise, providing consistent and supported interfaces for their security-relevant configuration, maintenance, and operation.
[HAMOD] O.HA_MANAGEMENT	To facilitate authorized management by the enterprise, conformant Host Agents provide consistent and supported interfaces for their security-relevant configuration and maintenance.
[App PP] 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.
[HAMOD] O.INTEGRITY	Conformant Host Agents will ensure the integrity of policy and/or commands sent to the Host Agent and also leverage execution environment or platform-based mitigations to protect the Host Agent.
[App PP] 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.
[App PP] 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.
[HAMOD] O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant Host Agents will use a trusted channel for sending and receiving data.
[App PP] 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.
[EDRMOD] O.PROTECTED_TRANSIT	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOE s will use a trusted channel to protect all communications. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application or to unauthenticated users.
[App PP] 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.

4.4.2 Security Objectives for the Operational Environment

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

Table 10: TOE Operational Environment Objectives

Objective	Objective Definition
[App PP] 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.
[App PP] 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.
[App PP] 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.
[EDRMOD] OE.RELIABLE_TRANSIT	Wired or wireless network traffic between the EDR and host agents will provide reasonably reliable connectivity.

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 Profile to which the TOE claims conformance.

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:** allows a component to be used more than once with varying operations. Indicated with a tag to represent the component in parentheses following the element number of the iterated SFR (e.g., "(Server)").

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.

If SFR text is reproduced verbatim from text that was formatted in a claimed PP (such as if the PP's instantiation of the SFR has a refinement or a completed assignment), the formatting is not preserved. This is so that the reader can identify the operations that are performed by the ST author as opposed to the PP author.

6.2 Security Functional Requirements Summary

The following table lists the SFRs claimed by the TOE:

Table 11: Security Functional Requirements for the Server

Class Name	Component Identification	Component Name
Security Audit	[EDRMOD] FAU_ALT_EXT.1	Server Alerts
	[EDRMOD] FAU_COL_EXT.1	Collected Endpoint Data
	[EDRMOD] FAU_GEN.1/EDR	Audit Data Generation
Cryptographic Support	[App PP] FCS_CKM_EXT.1(Server)	Cryptographic Key Generation Services
	[App PP] FCS_CKM.1/AK(Server)	Cryptographic Asymmetric Key Generation
	[App PP] FCS_CKM.2 (Server)	Cryptographic Key Establishment
	[App PP] FCS_RBG_EXT.1(Server)	Random Bit Generation Services
	[App PP] FCS_STO_EXT.1(Server)	Storage of Credentials
User Data Protection	[App PP] FDP_DAR_EXT.1	Encryption of Sensitive Application Data
	[App PP] FDP_DEC_EXT.1	Access to Platform Resources

	(Server)	
	[App PP] FDP_NET_EXT.1(Server)	Network Communications
Identification and Authentication	[EDRMOD] FIA_AUT_EXT.1	Dashboard Authentication Mechanisms
	[EDRMOD] FIA_PWD_EXT.1	Password Authentication
Security Management	[App PP] FMT_CFG_EXT.1	Secure by Default Configuration
	[App PP] FMT_MEC_EXT.1	Supported Configuration Mechanism
	[App PP] FMT_SMF.1 (Server)	Specification of Management Functions
	[EDRMOD] FMT_SMF.1/ENDPOINT	Specification of Management Functions (EDR Management of EDR)
	[EDRMOD] FMT_SMF.1/HOST	Specification of Management Functions (EDR Management of Host Agent)
	[EDRMOD] FMT_SMR.1	Security Management Roles
	[EDRMOD] FMT_SRF_EXT.1	Specification of Remediation Functions
Privacy	[App PP] FPR_ANO_EXT.1	User Consent for Transmission of Personally Identifiable Information
Protection of the TSF	[App PP] FPT_AEX_EXT.1	Anti-Exploitation Capabilities
	[App PP] FPT_API_EXT.1	Use of Supported Services and APIs
	[App PP] FPT_IDV_EXT.1	Software Identification and Versions
	[EDRMOD] FPT_ITT.1	Basic Internal TSF Data Transfer Protection
	[App PP] FPT_LIB_EXT.1	Use of Third Party Libraries
	[App PP] FPT_TUD_EXT.1 (Server)	Integrity for Installation and Update
	[App PP] FPT_TUD_EXT.2	Integrity for Installation and Update
Trusted Path/Channel	[App PP] FTP_DIT_EXT.1	Protection of Data in Transit
	[EDRMOD] FTP_TRP.1	Trusted Path

Table 12: Security Functional Requirements for the Sensor

Class Name	Component Identification	Component Name
Security Audit	[HAMOD] FAU_GEN.1/HA	Audit Data Generation
	[HAMOD] FAU_STO_EXT.1	Audit Data Storage
Cryptographic Support	[App PP] FCS_CKM_EXT.1(Sensor)	Cryptographic Key Generation Services
	[App PP] FCS_RBG_EXT.1(Sensor)	Random Bit Generation Services
	[App PP] FCS_STO_EXT.1(Sensor)	Storage of Credentials
User Data Protection	[App PP] FDP_DAR_EXT.1	Encryption of Sensitive Application Data

	[App PP] FDP_DEC_EXT.1(Sensor)	Access to Platform Resources
	[App PP] FDP_NET_EXT.1(Sensor)	Network Communications
	[HAMOD] FDP_NET_EXT.2	Network Communications
Host Agent	[HAMOD] FHA_CHA_EXT.1	Cache Host Agent Collected Data
	[HAMOD] FHA_COL_EXT.1	Collected Audit
	[HAMOD] FHA_HAD_EXT.1	Host Agent Declaration
	[App PP] FIA_X509_EXT.1	X.509 Certificate Validation
	[App PP] FIA_X509_EXT.2	X.509 Certificate Authentication
Security Management	[App PP] FMT_CFG_EXT.1	Secure by Default Configuration
	[App PP] FMT_MEC_EXT.1	Supported Configuration Mechanism
	[App PP] FMT_SMF.1(Sensor)	Specification of Management Functions
	[HAMOD] FMT_SMF.1/HA	Specification of Management Functions (Configuration of Host Agent)
	[HAMOD] FMT_UNR_EXT.1	User Unenrollment Prevention
Privacy	[App PP] FPR_ANO_EXT.1	User Consent for Transmission of Personally Identifiable Information
Protection of the TSF	[App PP] FPT_AEX_EXT.1	Anti-Exploitation Capabilities
	[App PP] FPT_API_EXT.1	Use of Supported Services and APIs
	[App PP] FPT_IDV_EXT.1	Software Identification and Versions
	[App PP] FPT_LIB_EXT.1	Use of Third Party Libraries
	[App PP] FPT_TUD_EXT.1(Sensor)	Integrity for Installation and Update
	[App PP] FPT_TUD_EXT.2	Integrity for Installation and Update
Trusted Path/Channel	[App PP] FTP_DIT_EXT.1	Protection of Data in Transit

6.3 Security Functional Requirements

6.3.1 Class FAU: Security Audit

6.3.1.1 *[EDRMOD] FAU_ALT_EXT.1 Server Alerts*

FAU_ALT_EXT.1.1[1]

The EDR shall alert authorized users on a management dashboard in the event of detection of potentially unauthorized activity on enrolled endpoints.

[1] TD0757

FAU_ALT_EXT.1.2

The EDR shall provide a visualization of detected alerts of potentially unauthorized incidents, and shall include:

a. An initial incident severity and [categorization],
b. An incident timeline.

FAU_ALT_EXT.1.3

The EDR shall provide a data export capability for selected alerts with a specified standards-based format of [

Log Event Extended Format (LEEF)

] .

6.3.1.2 *[EDRMOD] FAU_COL_EXT.1 Collected Endpoint Data*

FAU_COL_EXT.1.1[2]

The EDR shall collect the following minimum set of endpoint data from a Host Agent:

a. Operating System (OS) version, architecture, and IP Address,
b. Privileged and unprivileged endpoint account login activity,
c. Process creation,
d. Libraries and modules loaded by processes,
e. Filenames and [*no other metadata*] of files created and [*no other activities*] on persistent storage,
f. [*no other host data*].

6.3.1.3 *[EDRMOD] FAU_GEN.1/EDR Audit Data Generation*

FAU_GEN.1.1/EDR[3]

The EDR shall generate an audit record of the following auditable events:

b. All auditable events for the not specified level of audit;

and

a. EDR management dashboard log in activity;
b. Remediation commands sent to a Host Agent, affected endpoint, or network devices;
c. EDR configuration changes;
d. Change in Host Agent enrollment status,
e. [*No other auditable events*].

[2] TD0744
[3] TD0757

FAU_GEN.1.2/EDR

The EDR shall record within each audit record at least the following information:

a. Date and time of the event,
b. Type of event,
c. Subject identity,
d. Outcome (success or failure) of the event,
e. For each audit type, based on the auditable event definitions of the functional components included in the PP/ST, [*No other audit relevant information*].

6.3.1.4 *[HAMOD] FAU_GEN.1/HA Audit Data Generation*

FAU_GEN.1.1/HA

The Host Agent shall generate an audit record of the following auditable events:

b. All auditable events for the not specified level of audit;

and

a. Change in enrollment state with an ESM system,
b. [Generating] periodic heartbeat events,
c. [*no other auditable events*].

FAU_GEN.1.2/HA

The [Host Agent] shall record within each audit record at least the following information:

a. Date and time of the event,
b. Type of event,
c. Subject identity,
d. Outcome (success or failure) of the event,
e. For each audit type, based on the auditable event definitions of the functional components included in the PP/ST, [*no other audit information*].

6.3.1.5 *[HAMOD] FAU_STO_EXT.1 Audit Data Storage*

FAU_STO_EXT.1.1

The [Host Agent Platform] shall store audit events in the platform-provided logging mechanism.

6.3.2 Class FCS: Cryptographic Support

6.3.2.1 *[App PP] FCS_CKM_EXT.1(Server)[4] Cryptographic Key Generation Services*

FCS_CKM_EXT.1.1

The application shall [

[4] TD0717

- invoke platform-provided functionality for asymmetric key generation,

].

6.3.2.2 *[App PP] FCS_CKM_EXT.1(Sensor)[5] Cryptographic Key Generation Services*

FCS_CKM_EXT.1.1

The application shall [

- generate no asymmetric cryptographic keys,

].

6.3.2.3 *[App PP] FCS_CKM.1/AK(Server) Cryptographic Asymmetric Key Generation*

FCS_CKM.1.1/AK[6]

The application shall [

- invoke platform-provided functionality

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

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.1,
- ECC schemes using "NIST curves" P-384 and [no other curves] that meet the following: FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2,

].

6.3.2.4 *[App PP] FCS_CKM.2(Server) Cryptographic Key Establishment*

FCS_CKM.2.1[7]

The application shall [invoke platform-provided functionality] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"

].

[5] TD0717
[6] TD0717 and TD0945
[7] TD0717

6.3.2.5 *[App PP] FCS_RBG_EXT.1(Server)Random Bit Generation Services*

FCS_RBG_EXT.1.1

The application shall [

- invoke platform-provided DRBG functionality

] for its cryptographic operations.

6.3.2.6 *[App PP] FCS_RBG_EXT.1(Sensor)Random Bit Generation Services*

FCS_RBG_EXT.1.1

The application shall [

- use no DRBG functionality,

] for its cryptographic operations.

6.3.2.7 *[App PP] FCS_STO_EXT.1(Server) Storage of Credentials*

FCS_STO_EXT.1.1[8]

The application shall [

- invoke the functionality provided by the platform to securely store [*user credentials*]

] to non-volatile memory.

6.3.2.8 *[App PP] FCS_STO_EXT.1(Sensor) Storage of Credentials*

FCS_STO_EXT.1.1[9]

The application shall [

- invoke the functionality provided by the platform to securely store [*management server certificate chain, Sensor Group certificate*]

] to non-volatile memory.

6.3.3 Class FDP: User Data Protection

6.3.3.1 *[App PP] FDP_DAR_EXT.1 Encryption of Sensitive Application Data*

FDP_DAR_EXT.1.1

The application shall [

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

[8] TD0865
[9] TD0865

] in non-volatile memory.

6.3.3.2 *[App PP] FDP_DEC_EXT.1(Server)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 [

- no sensitive information repositories

].

6.3.3.3 *[App PP] FDP_DEC_EXT.1(Sensor)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*]

].

6.3.3.4 *[App PP] FDP_NET_EXT.1(Server)Network Communications*

FDP_NET_EXT.1.1

The application shall restrict network communication to [

- respond to [*management web interface request, host sensor check-in request*],

].

6.3.3.5 *[App PP] FDP_NET_EXT.1(Sensor)Network Communications*

FDP_NET_EXT.1.1

The application shall restrict network communication to [

- Respond to [*application initiated periodic management server polling*]

].

6.3.3.6 *[HAMOD] FDP_NET_EXT.2 Network Communications*

FDP_NET_EXT.2.1

The Host Agent shall restrict network communications to: [

- An ESM server,

]

6.3.4 Class FHA: Host Agent

6.3.4.1 *[HAMOD] FHA_CHA_EXT.1 Cache Host Agent Collected Data*

FHA_CHA_EXT.1.1[10]

The Host Agent shall cache collected data up to [*the Host Agent's configured Quota*] on [persistent storage] if the trusted channel is not available.

FHA_CHA_EXT.1.2[11]

The Host Agent shall [[*stop collecting and recording data*]] when the local cache is full.

6.3.4.2 *[HAMOD] FHA_COL_EXT.1 Collected Audit*

FHA_COL_EXT.1.1

The Host Agent shall collect the following minimum set of endpoint event data:

a. Operating System (OS) version, architecture, and IP Address,
b. Privileged and unprivileged endpoint account login activity,
c. Process creation,
d. Libraries and modules loaded by processes,
e. Network connection activity, including destination IP,
f. Files created on persistent storage,
g. [*no other host data*].

6.3.4.3 *[HAMOD] FHA_HAD_EXT.1 Host Agent Declaration*

FHA_HAD_EXT.1.1

The Host Agent shall operate with the following ESM Software: [

- Endpoint Detection and Response (EDR)

].

[10] TD0749
[11] TD0749

6.3.5 Class FIA: Identification and Authentication

6.3.5.1 *[EDRMOD] FIA_AUT_EXT.1 Dashboard Authentication Mechanisms*

FIA_AUT_EXT.1.1

The EDR shall [

- provide authentication based on username/password and [
  - no other factors

]

] to support logins to any management dashboard or API.

6.3.5.2 *[EDRMOD] FIA_PWD_EXT.1 Password Authentication*

FIA_PWD_EXT.1.1

The EDR shall support the following for the Password Authentication Factor:

1. Passwords shall be able to be composed of any combination of [upper and lower case letters], numbers, and special characters: ["!", "@", "#", "$", "%", "^", "&", "*", "(", ")"],
2. Password length up to [*64*] characters shall be supported.

6.3.5.3 *[App PP] FIA_X509_EXT.1(Sensor)X.509 Certificate Validation*

FIA_X509_EXT.1.1

The application shall [invoke platform-provided functionality] to validate certificates in accordance with the following rules:

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

6.3.5.4 *[App PP] FIA_X509_EXT.2(Sensor)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].

6.3.6 Class FMT: Security Management

6.3.6.1 *[App PP] FMT_CFG_EXT.1Secure 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.

6.3.6.2 *[App PP] FMT_MEC_EXT.1Supported Configuration Mechanism*

FMT_MEC_EXT.1.1

The application shall [

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

].

6.3.6.3 *[App PP] FMT_SMF.1(Server) Specification of Management Functions*

FMT_SMF.1.1

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

- [*Version check*]

].

6.3.6.4 *[App PP] FMT_SMF.1(Sensor) Specification of Management Functions*

FMT_SMF.1.1

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

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

].

6.3.6.5 *[EDRMOD] FMT_SMF.1/ENDPOINT Specification of Management Functions (EDR Management of EDR)*

FMT_SMF.1.1/ENDPOINT[12]

The EDR shall be capable of performing the following management functions:

Table 13: FMT_SMF.1.1/ENDPOINT Management Functions[13]

Management Function	Administrator	SOC Analyst	Read-Only User
Configure the amount of time to retain data collected by the EDR [*for 30 days*]	**X**	-	-
Obtain or display the connectivity status of a Host Agent	**X**	**X**	**X**
Define a configurable list of [file hashes]	**X**	**X**	-
Configure visual suppression of incident alerts based on a configurable list of [file hashes]	**X**	**X**	-

6.3.6.6 *[HAMOD] FMT_SMF.1/HA Specification of Management Functions (Configuration of Host Agent)*

FMT_SMF.1.1/HA

The Host Agent shall be capable of performing the following management functions:

[12] TD0776

[13] The Protection Profile denotes if a management function is Mandatory (M) or Optional (O) for a role to perform that function. The Security Target uses X to denote if a management function is able to be performed by that role on the TOE. This is to clarify where Optional selections have and have not been made.

Table 14: FMT_SMF.1.1/HA Management Functions[14]

Management Function	Administrator
Configure the frequency for sending Host Agent data to an ESM	X
Assign at least one label or tag to categorize individual endpoint systems	X

6.3.6.7 *[EDRMOD] FMT_SMF.1/HOST Specification of Management Functions (EDR Management of Host Agent)*

FMT_SMF.1.1/HOST[15]

The EDR shall be capable of performing the following functions that control behavior of the Host Agent:

Table 15: FMT_SMF.1.1/HOST Management Functions[16]

Management Function	Administrator	SOC Analyst	Read-Only User
Configure the frequency for sending Host Agent data to the EDR [*to be between 60 and 300 seconds*]	X	-	-
Assign a label or tag to categorize or group individual endpoint systems	X	X	-

6.3.6.8 *[EDRMOD] FMT_SMR.1 Security Management Roles*

FMT_SMR.1.1

The EDR shall maintain the roles administrator, SOC analyst, read-only user.

FMT_SMR.1.2

The EDR shall be able to associate users with roles.

6.3.6.9 *[EDRMOD] FMT_SRF_EXT.1 Specification of Remediation Functions*

FMT_SRF_EXT.1.1

The EDR shall be capable of performing the following remediation functions:

[14] The Protection Profile denotes if a management function is Mandatory (M) or Optional (O) for a role to perform that function. The Security Target uses X to denote if a management function is able to be performed by that role on the TOE. This is to clarify where Optional selections have and have not been made.
[15] TD0778
[16] The Protection Profile denotes if a management function is Mandatory (M) or Optional (O) for a role to perform that function. The Security Target uses X to denote if a management function is able to be performed by that role on the TOE. This is to clarify where Optional selections have and have not been made.

Table 16: FMT_SRF_EXT.1.1 Management Functions[17]

Management Function	Administrator	SOC Analyst	Read-Only User
Quarantine an endpoint by [logically quarantining the endpoint from the network unless allowlisted]	X	X	-
Terminate a running process on an endpoint	X	X	-
Retrieve potentially unauthorized or affected files from an endpoint	-	-	-

6.3.6.10 *[HAMOD] FMT_UNR_EXT.1 User Unenrollment Prevention*

FMT_UNR_EXT.1.1

The [Host Agent Platform] shall enforce a mechanism to prevent unprivileged users of the platform from unenrolling the Host Agent with the ESM system.

6.3.7 Class FPR: Privacy

6.3.7.1 *[App PP] 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,

].

6.3.8 Class FPT: Protection of the TSF

6.3.8.1 *[App PP] 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 [*pgsql-13 on the Server*].

FPT_AEX_EXT.1.2

The application shall [

- not allocate any memory region with both write and execute permissions

].

[17] The Protection Profile denotes if a management function is Mandatory (M) or Optional (O) for a role to perform that function. The Security Target uses X to denote if a management function is able to be performed by that role on the TOE. This is to clarify where Optional selections have and have not been made. Note that "Retrieve potentially unauthorized or affected files from an endpoint" is not being claimed for this evaluation.

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.

6.3.8.2 *[App PP] FPT_API_EXT.1 Use of Supported Services and APIs*

FPT_API_EXT.1.1

The application shall use only documented platform APIs.

6.3.8.3 *[App PP] FPT_IDV_EXT.1 Software Identification and Versions*

FPT_IDV_EXT.1.1

The application shall be versioned with [[*EDR Server .rpm: major release number, minor release number, patch number, build date number, build time number; EDR Server management UI: major release number, minor release number, patch number; EDR Host Sensor: major release number, minor release number, patch number, build date number*]].

6.3.8.4 *[EDRMOD] FPT_ITT.1 Basic Internal TSF Data Transfer Protection*

FPT_ITT.1.1

The EDR shall [

- invoke platform-provided functionality for [HTTPS]

] to protect TSF data from modification, disclosure when it is transmitted between separate parts of the TOE.

6.3.8.5 *[App PP] FPT_LIB_EXT.1 Use of Third Party Libraries*

FPT_LIB_EXT.1.1

The application shall be packaged with only [*third-party libraries listed in Table 24*].

6.3.8.6 *[App PP] FPT_TUD_EXT.1(Server) Integrity for Installation and Update*

FPT_TUD_EXT.1.1

The application shall [leverage the platform] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2

The application shall [provide the ability, 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

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].

6.3.8.7 *[App PP] FPT_TUD_EXT.1(Sensor)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

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].

6.3.8.8 *[App PP] FPT_TUD_EXT.2 Integrity for Installation and Update*

FPT_TUD_EXT.2.1[18]

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

FPT_TUD_EXT.2.2

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

FPT_TUD_EXT.2.3

[18] TD0628

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

6.3.9 Class FTP: Trusted Path/Channel

6.3.9.1 ***[App PP] FTP_DIT_EXT.1*** ***Protection of Data in Transit***

FTP_DIT_EXT.1.1[19]

The application shall [

- not transmit any [data]

] between itself and another trusted IT product.

6.3.9.2 ***[EDRMOD] FTP_TRP.1*** ***Trusted Path***

FTP_TRP.1.1

The EDR shall [

- invoke platform-provided functionality for [HTTPS]

] to provide a communication path between itself and remote administrators that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification, disclosure.

FTP_TRP.1.2

The EDR shall [implement functionality] to permit remote administrators to initiate communication via the trusted path.

FTP_TRP.1.3

The EDR shall [implement functionality] to require the use of the trusted path for all remote administration actions.

6.4 Statement of Security Functional Requirements Consistency

The Security Functional Requirements included in the ST represent all required SFRs specified in the claimed PP as well as a subset of the selection based and objective 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.

7 Security Assurance Requirements

This section identifies the Security Assurance Requirements (SARs) that are claimed for the TOE. The

[19] TD0743

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)
	Timely Security Updates (ALC_TSU_EXT.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

8.1 Security Audit

8.1.1 [EDRMOD] FAU_ALT_EXT.1

The TOE alerts for potentially unauthorized activity on enrolled endpoints through utilizing customizable watchlists and threat intelligence feed-based detections that flag unauthorized activity types. The unauthorized activity types are set by an Administrator through the creation of custom Watchlists, Process Searches, Binary Searches, or Windows Event Log searches. Unauthorized activity types are also set by utilizing the Threat Intelligence Feeds page in the Web GUI. The Threat Intelligence feeds subscribe to specific activity types, and the TOE alerts the user according to those subscriptions. The threat intelligence feeds are sourced both internally to Carbon Black and externally through third-party Carbon Black partners and integrations.

Alerts are displayed through the Triage Alerts page of the TOE's Web GUI. Every alert generated by the TOE contains the computer name, domain name and IP address, status, last time the sensor was active, OE version, tamper protection level, server certificate, node ID, and sensor version. Detected alerts are assigned an initial incident severity, which is reported as a score from 0-100, and are categorized according to the report type that is included within the alert. All detected alerts include a timestamp and if applicable, include a timeline of events. The TOE provides the Administrator the ability to export selected alert data in the Log Event Extended Format (LEEF) through the Web GUI.

8.1.2 [EDRMOD] FAU_COL_EXT.1

The TOE supports the collection of data from the Endpoint System the Host Sensor is installed upon. The Host Sensor collects the data and sends that data to the EDR Server. The TOE will always collect the Operating System version, architecture, and IP address of the Endpoint System the Host Sensor is installed on. Other data collected is based upon the configuration of the TOE which will include the collection of successful login, failed login, and escalated privilege login data from a Host Sensor for all OS users. Data regarding process creation, as well as the libraries and modules loaded by processes, are also collected from the Endpoint System the Host Sensor is installed on. Filenames of files created on the Endpoint System's persistent storage are collected by the TOE.

8.1.3 [EDRMOD] FAU_GEN.1/EDR

The EDR Server generates audit records for the following auditable events:

- User authentication to the EDR Server's Web GUI
- Remediation commands sent to the Host Sensor – isolating a host and hash banning (Described in FMT_SRF_EXT.1)
- EDR configuration changes by users through the Web GUI
- Change in Host Sensor enrollment status

Audit records of log-in activity for EDR Server consist of successful and unsuccessful authentication attempts made to the EDR Server remotely through the Web GUI. Details regarding the remediation commands that are sent to a Host Sensor as well as EDR Server configuration changes are captured in the "requested information" portion of the audit records. Changes in Host Sensor enrollment status are

recorded on the Sensor page in the Web GUI.

All audit records produced by the TOE include date (dd/mon/yyyy) and time (hh:mm:ss) of the event, type of event, subject identity, and outcome (success or failure) of the event. No other audit relevant information is recorded within the audit records.

8.1.4 [HAMOD] FAU_GEN.1/HA

The TOE generates audit records for events that occur within the TSF. The Host Sensor generates audit records for all administrative functions that occur on the Host Sensor including:

- The change in its enrollment status within the ESM system
- The generation of periodic heartbeat events for connection attempts to the EDR Server

All audit records produced by the TOE include date (mm/dd/yyyy) and time (hh:mm:ss) of the event, type of event, username of the subject who triggered the event, and success or failure of the event. No other audit relevant information is recorded within the audit records.

8.1.5 [HAMOD] FAU_STO_EXT.1

The Host Sensor sends all audit records to the underlying operating system's Windows Event Log. No audit records are stored within the Host Sensor itself.

8.2 Cryptographic Support

8.2.1 [App PP] FCS_CKM_EXT.1(Server), FCS_CKM_EXT.1(Sensor), and FCS_CKM.1/AK

[EDR Server] The EDR Server invokes the operating system for establishing HTTPS/TLS communication (FPT_ITT.1 and FTP_TRP.1) and relies on the operating system's OpenSSL cryptographic module to generate the Elliptic curve-based keys in support of the Elliptic curve-based key establishment.

The EDR Server invokes the operating system's OpenSSL cryptographic module to generate RSA scheme certificates for the Sensor Group Certificate. These certificates are used for Host Sensor Group identification during the check-in cycles. All members of a Sensor Group use the same Sensor Group certificate for group identification.

The following are the key scheme specifications:

- ECC schemes using "NIST curves" P-384 and no other curves that meet the following: FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix B.4. ECC schemes are used for supporting HTTPS/TLS communication (FPT_ITT.1 and FTP_TRP.1)
- RSA schemes using cryptographic key sizes of 2048-bit that meet the following FIPS PUB 186-5, "Digital Signature Standard (DSS), Appendix B.3.

[Host Sensor] The Host Sensor does not perform any asymmetric key generation. Sensor Group certificates for HTTPS/TLS communication are imported as part of the Host Sensor's initial installation and are generated on the EDR Server.

8.2.2 [App PP] FCS_CKM.2(Server)

The EDR Server invokes the platform to use Elliptic curve-based key establishment scheme for the establishment of HTTPS/TLS communications. This key establishment scheme supports the usage of a Web GUI for remote administration of the TOE and EDR Server to Host Sensor platform communications. Elliptic curve-based key establishment conforms to NIST SP 800-56A.

NOTE: The RSA scheme certificates generated under FCS_CKM_EXT.1 are for Host Sensor Group identification during the check-in cycles. The Sensor Group certificate is not used for HTTPS/TLS communication establishment and is therefore not claimed for key establishment.

8.2.3 [App PP] FCS_RBG_EXT.1(Server) and FCS_RBG_EXT.1(Sensor)

[EDR Server] The EDR Server invokes the platform for key generation and trusted communications; therefore, the platform calls on the DRBG services required. TOE invokes the platform to use AES counter DRBG seeded with 256 bits of entropy.

From Python, the Server calls os.urandom which calls OS getrandom() in blocking mode which pulls entropy from /dev/urandom to:

- Setup seed for pbkdf2_hmac encryption algorithm used for hashing user passwords in support of FDP_DAR_EXT.1 Encryption of Sensitive Application Data.

The EDR Server provides no ability to specify the use of an alternative DRBG.

[Host Sensor] The Host Sensor invokes the Windows platform for encrypted storage of credentials and trusted communications. Therefore, the Host Sensor does not call on DRBG services but rather the Windows platform calls on the DRBG services required.

8.2.4 [App PP] FCS_STO_EXT.1(Server) and FCS_STO_EXT.1(Sensor)

[EDR Server] The EDR Server invokes the underlying platform to provide the hash of the user password before storing the credentials in the internal database. At no time is a password stored as plain text. The EDR Server requires the underlying platform to provide data-at-rest encryption using LUKS encrypted partitioning to protect the internal database.

The EDR Server does not invoke or implement the storage of keys. CA and server keys are manually created outside of the EDR Server and must be manually stored on the platform per RHEL guidance. As part of the pre-requisite installation procedures, the certificate and key must be imported and validated. Upon successful validation the certificate and key are then copied onto the server in /etc/certs/cb-server.crt and /etc/certs/cb-server.key and becomes the default certificate.

[Host Sensor] 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.

Table 17: Certificate and Key Store

Key Material	Purpose	Storage Location	Origin
Management Server Certificate (public key)	Used by the host platform OS to validate the management server	Microsoft Key Storage Provider; located in LocalMachine\CarbonBlack certificate store.	Pre-generated by customer and installed on management server. The management server

	during the HTTPS/TLS channel establishment		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.

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.

8.3 User Data Protection

8.3.1 [App PP] FDP_DAR_EXT.1

[EDR Server] The EDR Server defines the following sensitive data objects:

- Sensor Group Certificates: identification of the Host Sensor's group used during the check-in cycles with TOE. All members of a Sensor Group use the same Sensor Group certificate for group identification. This identifier allows the TOE to know how to classify the source of information received from the Host Sensor and what Host Sensor policies the TOE must provide to the Host Sensor during the check-in cycles. These are not used for HTTPS/TLS communication establishment.
- User credentials: usernames and the hashed representation of the passwords are stored in accordance with FCS_STO_EXT.1.

Sensitive data is stored in the EDR Server's internal database. As a result of the OS platform being Linux based, the EDR Server requires the underlying platform to provide data-at-rest encryption using LUKS encrypted partitioning to protect the internal database.

Instructions for preparing the Operational Environment, including the requirement for LUKS, are provided in the supplemental administrative guidance that is included with the TOE.

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

There is no interactive user interface provided by the Host Sensor and therefore there is no user data generation possible from the Host Sensor. The Host Sensor itself records 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 Host Sensor 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.

8.3.2 [App PP] FDP_DEC_EXT.1(Server) and FDP_DEC_EXT.1(Sensor)

[EDR Server] The EDR Server relies on its underlying platform to provide network connectivity in order to establish communications channels. There are no sensitive information repositories (storage locations for private user or Administrator data) that the EDR Server requires access to.

[Host Sensor] The Host Sensor restricts its access to the Endpoint System's network connectivity hardware resource for the purposed defined in FDP_NET_EXT.1.

The EDR Sensor gathers information from system repositories based on the collection policy administratively configured at the EDR Server. The collected information is transmitted to the EDR Server for monitoring and alerting functionality. The only Endpoint System's sensitive information that the Host Sensor 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.

8.3.3 [App PP] FDP_NET_EXT.1(Server) and FDP_NET_EXT.1(Sensor)

[EDR Server] The EDR Server requires the RHEL 8.6 OS platform to initiate a HTTPS/TLS (TLS v1.2) server process that only supports the following two remote connections requests:

- Response to requests from a remote Administrator using a web browser to access the TOE's Web GUI. Port for remote administration: 443.
- Response to the Host Sensor platform's requests for a HTTPS/TLS connection for the periodic check-in. If the HTTPS/TLS connection is interrupted, the Host Sensor platform will automatically attempt to re-establish the connection. Port for Host Sensor communication: 443 by default, but can be configured by the Administrator.

  During a check-in,

  - The EDR Server receives 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 from the Host Sensor.

- The Host Sensor pulls configuration and data collection policy updates, Sensor Group change information including new Sensor Group certificate (if applicable), notification of Host Sensor software update being available, and Host Sensor update package (if applicable).

The EDR Server does not provide client functionality and does not initiate external network connections to another device.

[Host Sensor] The Host Sensor requires network access to communicate with the EDR Server. During startup the Host Sensor invokes the platform to establish a persistent HTTPS/TLS connection to the EDR 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 Host Sensor:

- 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 acceptlist/denylist definitions, Sensor Group change, and TOE software updates.

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

8.3.4 [HAMOD] FDP_NET_EXT.2

The Host Sensor restricts its network communications to only the EDR Server (i.e., an ESM server). If a deployment of the TOE includes more than one Host Sensor, there is no agent-to-agent communication occurs between the Host Sensors.

8.4 Host Agent

8.4.1 [HAMOD] FHA_CHA_EXT.1

The Host Sensor stores its collected data in one or more persistent event files on its underlying platform. An initial event file is created upon installation of the Host Sensor on the Endpoint System. Additional event files are automatically created when the event file in use by the TOE reaches its capacity of 5MB. New event files are created until the collective size of the event files meets the Quota limit. The Quota limit is set by an Administrator through the EDR Server's Web GUI, and the setting is sent to the Host Sensor during reachability events.

When a connection from the Host Sensor is established to the EDR Server, the Host Sensor will send each event file's data to the EDR Server one at a time. This means that the EDR Server will send a response to the Host Sensor each time it has received an event file's data. Once the Host Sensor receives this confirmation from the EDR Server, the associated event file is deleted from the local storage on the Host Sensor's underlying platform. This process is repeated for each event file in the local storage on the Host Sensor's underlying platform that has accumulated since the last connection was made between the Host Sensor and the EDR Server.

If the connection between the Host Sensor and the EDR Server is lost, the data collected from the Host

Sensor remains stored in the local event files. In the event that the Quota limit is reached during a connection outage, no additional information is collected or recorded to the event files. Once the connection is restored between the Host Sensor and EDR Server, the process of sending collected data to the EDR Server resumes.

8.4.2 [HAMOD] FHA_COL_EXT.1

The Host Sensor supports the collection of the Operating System version, architecture, and IP address of the Endpoint System the Host Sensor is installed on. Privileged and unprivileged endpoint user account authentication activity is collected, to include login success and login failure for initial authentication to the Operating System as well as elevated privilege authentication. Data regarding process creation as well as the libraries and modules loaded by processes is also collected from the Endpoint System. The Host Sensor also collects data regarding the Endpoint System's network connection activity including destination IPs. Files created on the Endpoint System's persistent storage are collected.

8.4.3 [HAMOD] FHA_HAD_EXT.1

The Host Sensor operates with the EDR Server and no other ESM software.

8.5 Identification and Authentication

8.5.1 [EDRMOD] FIA_AUT_EXT.1

The EDR Server provides username/password based authentication to support user access to the Web GUI, the TOE's management dashboard. The EDR Server has its own user store within its internal database. The credentials stored within this internal database are used to verify authentication attempts to the Web GUI.

8.5.2 [EDRMOD] FIA_PWD_EXT.1

In the evaluated configuration, the TOE supports passwords containing up to 64 characters. Passwords can be composed of any combination of upper and lower-case Basic Latin letters, numbers and special characters. The accepted special characters include: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")".

8.5.3 [App PP] FIA_X509_EXT.1 and FIA_X509_EXT.2

The Host Sensor validates the presented X.509v3 server certificate chain, during the TLS handshake, to support the establishment of HTTPS/TLS non-mutual authenticated connections between the EDR Server and Host Sensor in accordance with RFC 5280. The application determines the validity of certificates by ensuring that certificate path validation occurs and that the certificate path is valid in accordance with RFC 5280. In addition:

- The application treats a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE
- The certificate path must terminate with the trusted CA certificate that was packaged with the installation instance that was pushed to and installed on the host endpoint system.
- The application validates a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates, and that any constraints

are met.

- If the certificate is deemed invalid, the certificate is rejected and denies the connection. If the certificate is deemed valid the application validates the certificate revocation status using a OCSP as specified in RFC 6960. The TOE does not provide any administrative override functionality.
- When the application cannot establish a connection to determine the validity of a certificate the TSF does not accept the certificate and denies the connection.
- The application validates the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification must 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 must 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 must 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 must have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) 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.

8.6 Security Management

8.6.1 [App PP] FMT_CFG_EXT.1

[EDR Server] The EDR Server requires the Administrator to input valid credentials before being granted access to administer the TOE via the Web GUI. Upon initial installation of the EDR Server, the Administrator is prompted to create a username and password. There are no default credentials available for initial authentication to the EDR Server. The EDR Server software is automatically installed with the appropriate file permissions to prevent unauthorized access to the binaries, configuration settings, and data.

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

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

8.6.2 [App PP] FMT_MEC_EXT.1

[EDR Server] The EDR Server relies on the underlying platform's recommended methods for storing and setting configuration options. The TOE has one configuration file, /etc/cb/cb.conf, that contains a parameter for setting the retention time for retaining collected EDR data (SolrAuthCorePurgeDays) and another for setting the frequency for sending Host Agent data to an ESM (SensorCheckinDelaySec). These parameters can only be changed by an OS administrator which is considered one of the security administrators. All other parameters are managed through the TOE's Web GUI.

[Host Sensor] All configuration settings, as established by the EDR Server, are stored on the Host Sensor according to the Windows platform best practices. The credential data is stored according to FCS_STO_EXT.1. Additionally, the Host Sensor application uses the Windows Registry (HKLM/Software/CarbonBlack) to store application configuration settings.

The Host Sensor 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 Host Sensor as a service by the OS. There are no Host Sensor 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 Host Sensor 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 Host Sensor 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 Host Sensor configuration parameters under this registry key.

8.6.3 [App PP] FMT_SMF.1(Server) and FMT_SMF.1(Sensor)

[EDR Server] The EDR Server provides a Web GUI to allow Administrators to remotely manage the TSF and management of the server application itself. Only the functions that are relevant to the TSF and have SFR claims are discussed in this Security Target. The following security-relevant management functions are provided by the EDR Server:

- enable/disable the transmission of any information describing the system's hardware, software, or configuration,

- Ability to query the software version (FPT_TUD_EXT.1).

There is no management function available to check for TOE updates. The actual downloading and installation of any downloaded updates is manually performed by a person with root access to the platform and therefore is not included in this list.

[Host Sensor] Both the Local Administrator and Endpoint User are considered the owner or user of the endpoint device for which the Host Sensor is installed. An EDR server administrator can enable the Local Administrator (Windows OS administrator) the ability to 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. However, a typical endpoint user does not have any management functionality.

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 Host Sensor during 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 Host Sensor are accomplished using the Window interfaces such as Windows Services (start/stop TOE), appwiz.cpl (uninstall TOE), or invoked by the management server.

As stated above, the Administrator can modify the data collection policy from the EDR Server which controls what system data (enable/disable) is collected and ultimately transmitted to the EDR Server. Upon receiving the configuration change, the Host Sensor 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.

8.6.4 [EDRMOD] FMT_SMF.1/ENDPOINT

The following TSF management functions are restricted to the associated user types defined by Table 18:

Table 18: FMT_SMF.1/ENDPOINT Management Functions

Management Function	User Type Permitted to Perform Management Function
Configure the amount of time to retain data collected by the EDR for 30 days	Administrator (OS Administrator on EDR Server)
Obtain or display the connectivity status of a Host Agent	Administrator (Global Admin), SOC Analyst (Analyst), Read-Only (Viewer)
Define a configurable list of file hashes	Administrator (Global Admin), SOC Analyst (Analyst)
Configure ingress filter/banned hash to visual suppression of incident alerts based on a configurable list of file hashes	Administrator (Global Admin), SOC Analyst (Analyst)
Configure state filter to visual suppression of incident alerts based on a configurable list of file hashes (watchlists)	Administrator (Global Admin), SOC Analyst (Analyst)

The amount of time that the EDR Server will retain data is configured through a configuration file located on the EDR Server system. Only an Administrator with access to the EDR Server's Operating System and 'sudo' privileges has the ability to configure the retention time frame. The Host Sensor connectivity status (i.e., online, offline, or uninstalled) is displayed in the Web GUI at the top of the Sensor Details page for each Host Sensor.

The TOE utilizes ingress filtering to visually suppress alerts based on a configurable list of file hashes

that are user definable within the Web GUI. The EDR Server checks the MD5 hashes of process executables against the list of file hashes defined in the filter to remove the data upon ingress to the EDR Server and therefore, preventing an alert from ever being displayed.

8.6.5 [HAMOD] FMT_SMF.1/HA

The following TSF management functions on the Host Sensor are restricted to the associated user types defined by Table 19:

Table 19: FMT_SMF.1/ENDPOINT Management Functions

Management Function	User Type Permitted to Perform Management Function
Configure the frequency for sending Host Agent data to an ESM	Administrator (OS Administrator on EDR Server)
Assign at least one label or tag to categorize individual endpoint systems	Administrator (Administrator (Global Admin), SOC Analyst (Analyst)

The frequency for sending Host Sensor data to the EDR Server is specified in a configuration file on the EDR Sensor's platform. Only an Administrator with access to the EDR Server's Operating System and 'sudo' privileges has the ability to configure the frequency setting. The 'SensorCheckinDelaySec' parameter is used to set the frequency and can be set between 60 seconds and 300 seconds.

The Host Sensors are categorized by Sensor Group which is initially assigned upon installation of the TOE but can also be changed post-installation. All Host Sensors are part of either the default Sensor Group or a custom Sensor Group. Sensor Group management is performed by the EDR Server, which is the authorized Administrator of the Host Sensor.

8.6.6 [EDRMOD] FMT_SMF.1/HOST

The following TSF management functions are restricted to the associated user types defined by Table 20:

Table 20: FMT_SMF.1/HOST Management Functions

Management Function	User Type Permitted to Perform Management Function
Configure the frequency for sending Host Agent data to the EDR to be between 60 and 300 seconds	Administrator (OS Administrator on EDR Server)
Assign a label or tag to categorize or group individual endpoint systems	Administrator (Global Admin), SOC Analyst (Analyst)

The frequency for sending Host Sensor data to the EDR Server is defined in a configuration file on the EDR Server's platform. Only an Administrator with access to the EDR Server's Operating System and 'sudo' privileges has the ability to configure the frequency setting. Upon the Host Sensor's next check-in, the EDR Server will send the new frequency value to the Host Sensor for saving in the Host Sensor's local configuration file.

Host Sensors are categorized by Sensor Group which is initially assigned upon installation of the TOE but can also be changed post-installation. All Host Sensors are part of either the default Sensor Group or a custom Sensor Group. Sensor Group management is executed through the Web GUI on the EDR Server. Upon the Host Sensor's next check-in, the EDR Server will send information related to the newly assigned Sensor Group to the Host Sensor for saving in the Host Sensor's local configuration file.

8.6.7 [EDRMOD] FMT_SMR.1

The TOE supports three types of user roles. The types of user roles and the permissions associated with each role are as follows:

- Global Admin – Maps to the "Administrator" role defined by the PP. The Global Admin has the ability to perform all PP defined management functions.
- Analyst – Maps to the "SOC Analyst" role defined by the PP. The Analyst role allows the user to monitor and respond to suspicious or malicious activity on endpoints in sensors groups for which it has the role.
- Viewer – Maps to the "Read-Only" role defined by the PP. The Viewer role allows the user to view information, including suspicious or malicious activity, on endpoints in Sensor Groups for which it has the role.

The TOE is designed to use permissions which allow, limit or prevent user access to specific administrative tools based on the aligned user role. Upon successful authentication, the TSF associates the administratively defined set of permissions (role) for that user to the subject acting on behalf of that user. The TSF then enforces role-based access control (RBAC) to limit access to TSF functions and data based on the set of permissions bound to the subject.

8.6.8 [EDRMOD] FMT_SRF_EXT.1

The TOE, through the EDR Server, provides the capability to administer remediation functions in the event of suspected malicious activities occurring on the Endpoint System through logical quarantining and hash banning.

The TOE has the capability to logically quarantine an endpoint from the network, unless the endpoint has been placed on an allowlist to allow specific connections to be made to those endpoints. An Administrator or SOC Analyst initiates the quarantine process by changing the status of the endpoint to isolated through the Web GUI. A previously isolated endpoint can be unquarantined if the Administrator or SOC Analyst changes the Endpoint System's status to unisolated. The change in status from isolated to unisolated and vice versa does not occur automatically and must be initiated by the Administrator or SOC Analyst user.

The TOE provides the capability to terminate a currently running process on an Endpoint System. The process ban functionality uses a denylist of hashes for known process and will terminate any running process which matches a hash on the denylist. When an Administrator or SOC Analyst identifies an unwanted process, they add the associated MD5 hash to the Banned Hash page (i.e., denylist) within the Web GUI. The process ban must be confirmed within the Sensor Group configuration page before the ban takes effect.

8.6.9 [HAMOD] FMT_UNR_EXT.1

The unenrolling process is protected by Windows permissions that constrain users' ability to uninstall applications. Only an Administrator with access to the Host Sensor's platform and Windows administrative privileges has the ability to unenroll a Host Sensor at the platform. The Host Sensor platform's Windows Application Management performs the unenrolling of the Host Sensor by uninstalling the application. The unenrollment process is self-contained to the Host Sensor as there is no action needed on the EDR Server side to complete this process. An EDR Administrator may uninstall as senor using the EDR management UI via the Sensors>Actions>uninstall Sensor function.

8.7 Privacy

8.7.1 [App PP] FPR_ANO_EXT.1

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

8.8 Protection of the TSF

8.8.1 [App PP] FPT_AEX_EXT.1

[EDR Server] The EDR Server implements several mechanisms to protect against exploitation. The application does not map memory to an explicit address with the exception of the pgsql-13 (PostgreSQL package included with the TOE). PostgreSQL allocates and manages memory chunks for private PostgreSQL usage for out of memory prevention. Additionally, the EDR server does not allocate any memory region with write and execute permissions, nor does it write user-modifiable files to directories that contain executable files.

The application is compatible with the platform-provided security feature SELinux enabled and enforcing. The EDR Server provides its own copy of Python compiled with -fPIE flags. The python-based EDR Server subcomponents are also compiled with -fPIE flags to ensure ASLR enforcement. The Java-based EDR Server subcomponents rely on the default Java virtual machine that comes with a RHEL distribution.

[Host Sensor] The Host Sensor is a Windows service and does not operate in user space meaning the application does not provide a user interface for users to generate user-modifiable files. The Host Sensor application does not map memory to an explicit address or allocate any memory region with both write and execute permissions. The Host Sensor is capable of operating with Windows Defender Exploit Guard Exploit Protection configured given the following minimum mitigations are enabled;

- Control Flow Guard (CFG)
- Randomize memory allocations (Bottom-Up ASLR)
- Export address filtering (EAF)
- Import address filtering (IAF)
- 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.

Table 21: Compiler Flags used by the TOE

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

8.8.2 [App PP] FPT_API_EXT.1

[EDR Server] When the EDR Server is installed on the RHEL OS, it uses only the following supported platform APIs in order to function.

Table 22: EDR Server APIs

Java Framework API:		
javax.naming.ldap.LdapName	java.net.InetSocketAddress	java.sql
javax.naming.ldap.Rdn	java.net.MalformedURLException	java.text
javax.servlet.	java.net.ServerSocket	java.time
javax.xml.bind.DatatypeConverter	java.net.Socket	java.util
java.io	java.net.UnknownHostException	java.util.zip
java.lang.annotation	java.net.URI	javax.naming.ldap
java.lang.invoke.MethodHandles	java.net.URL	javax.servlet
java.lang.management	java.nio.BufferUnderflowException	javax.xml.bind.DatatypeConverter
java.lang.reflect	java.nio.file	org.slf4j.Logger
java.lang.Thread	java.security.cert.CertificateFactory	org.slf4j.LoggerFactory
java.net.InetAddress	java.security.cert.X509Certificate	
Python Framework API:		
abc	grp	pytz
antlr4	gzip	queue
argparse	hashlib	random
ast	heapq	rapidjson
atexit	http	re
base64	imp	requests
binascii	importlib	saml2

bisect	inspect	select
boto3	io	selinux
bs4	ipaddress	setuptools
builtins	itertools	shlex
cprofile	json	shutil
calendar	kombu	signal
code	locale	simplejson
codecs	logging	smtplib
collections	markdown	socket
configparser	math	stackcollector
contextlib	migrate	stat
copy	mmap	struct
coverage	multiprocessing	subprocess
crypto	netaddr	supervisor
csv	numpy	sys
ctypes	openssl	tabulate
difflib	operator	tarfile
distutils	optparse	tempfile
email	os	termcolor
enum	pathlib	textwrap
errno	pickle	threading
fcntl	pid	traceback
fileinput	pkg_resources	urllib
fnmatch	pkgutil	urllib3
functools	platform	warnings
gc	pprint	weakref
getpass	pstats	werkzeug
gevent	psutil	xml
glob	psycopg2	xmlrpc
greenlet	pwd	zipfile
ghrequests	pyformance	zlib

[Host Sensor] When the Host Sensor is installed on the Windows OS, it uses only the following supported Windows platform APIs in order to function.

Table 23: EDR Windows Sensor DLL files

Windows DLL containing system calls used by TOE	
advapi32.dll	rasadhlp.dll
amsi.dll	rpcrt4.dll
bcrypt.dll	RPCRT4.dll
bcryptprimitives.dll	rpcss.dll
ci.dll	rsaenh.dll
clbcatq.dll	schannel.dll
combase.dll	sechost.dll
coml2.dll	SECHOST.dll
crypt32.dll	secur32.dll
cryptbase.dll	setupapi.dll
cryptnet.dll	SHCore.dll

cryptsp.dll	shell32.dll
dbghelp.dll	shlwapi.dll
dhcpcsvc6.dll	SHLWAPI.dll
dhcpcsvc.dll	sspicli.dll
dnsapi.dll	tdh.dll
dpapi.dll	tzres.dll
fastprox.dll	ucrtbase.dll
fltLib.dll	user32.dll
fveui.dll	USER32.dll
gdi32.dll	userenv.dll
GDI32.dll	version.dll
gdi32full.dll	wbemcomn.dll
GdiPlus.dll	wbemprox.dll
gpapi.dll	wbemsvc.dll
IconCodecService.dll	webio.dll
imagehlp.dll	win32u.dll
imm32.dll	WindowsCodecs.dll
kernel32.dll	windows.storage.dll
kernel.appcore.dll	winhttp.dll
KernelBase.dll	winnlsres.dll
MpOAV.dll	winnsi.dll
msasn1.dll	wintrust.dll
msctf.dll	ws2_32.dll
msvcp_win.dll	wshqos.dll
msvcrt.dll	wtsapi32.dll
mswsock.dll	wuaueng.dll
ncrypt.dll	ntmarta.dll
ncryptprov.dll	ole32.dll
ncryptsslp.dll	oleaut32.dll
ngcrecovery.dll	OLEAUT32.dll
nsi.dll	profapi.dll
ntasn1.dll	psapi.dll
ntdll.dll	

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)

8.8.3 [App PP] FPT_IDV_EXT.1

[EDR Server] The TOE is versioned using a "major.minor.patch.builddate.buildtime" methodology for using the yum info cb-enterprise command. 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. The build date and time reflect the timestamp when compilation occurred. The TOE's Web GUI also displays the version in the bottom right corner of the web page. This format follows a "major.minor.patch" scheme.

[Host Sensor] To verify the Host Sensor version, the Security administrator would check the Sensor Page on the EDR Management UI. The reported version will be in the "major.minor.patch.builddate" format. Optionally, the version can be checked via the Windows application wizard by scrolling the list for Carbon Black Sensor. The reported version will be in the "major.minor.patch.builddate" format.

8.8.4 [EDRMOD] FPT_ITT.1

The Host Sensor invokes the underlying platform to establish an HTTPS connection with TLS v1.2 to protect the TSF data when it is in transit between the EDR Server and the Host Sensor. The Host Sensor always initiates the HTTPS connection to the EDR Server, and the connection remains open for the entirety of the reachability event. There are two types of reachability events that establish this connection:

- Periodic event – Host Sensor performs a check-in (i.e., heartbeat) with the EDR Server. Check-in occurs every 30 seconds by default. The check-in frequency can be changed by an Administrator through a configuration file on the EDR Server. Upon the Host Sensor's next check-in, the EDR Server will send the new frequency value to the Host Sensor for saving in the Host Sensor's local configuration file.
- Tamper event – Host Sensor initiates this connection when a possible threat is detected.

8.8.5 [App PP] FPT_LIB_EXT.1

[EDR Server] The EDR Server is packaged with the following third-party open source libraries.

Table 24: EDR Server Dynamic Libraries

Package Directory	Dynamic Library

temurin-17 /usr/lib/jvm/temurin-17-jre/lib/	libawt.so
	libawt_headless.so
	libawt_xawt.so
	libdt_socket.so
	libextnet.so
	libfontmanager.so
	libinstrument.so
	libj2gss.so
	libj2pcsc.so
	libj2pkcs11.so
	libjaas.so
	libjava.so
	libjavajpeg.so
	libjawt.so
	libjdwp.so
	libjimage.so
	libjli.so
	libjsig.so
	libjsound.so
	libjsvml.so
	liblcms.so
	libmanagement.so
	libmanagement_agent.so
	libmanagement_ext.so
	libmlib_image.so
	libnet.so
	libnio.so
	libprefs.so
	librmi.so
	libsctp.so
	libsplashscreen.so
	libsyslookup.so
	libverify.so
	libzip.so
	server/libjsig.so
	server/libjvm.so
Erlang /usr/lib64/erlang/lib/	asn1-5.2.2/priv/lib/asn1rt_nif.so
	crypto-5.4.2.3/priv/lib/crypto.so
	crypto-5.4.2.3/priv/lib/crypto_callback.so
	crypto-5.4.2.3/priv/lib/otp_test_engine.so

	runtime_tools-2.0.1/priv/lib/dyntrace.so
	runtime_tools-2.0.1/priv/lib/trace_file_drv.so
	runtime_tools-2.0.1/priv/lib/trace_ip_drv.so
libsensors /usr/lib64/	libsensors.so.4 lm_sensors-libs
	libsensors.so.4.4.0 lm_sensors-libs
jemalloc /usr/lib64/	libjemalloc.so.2
Python3 /usr/share/cb/virtualenv/lib/python3.10/site-packages/	Crypto/Cipher/_raw_aes.abi3.so
	Crypto/Cipher/_pkcs1_decode.abi3.so
	Crypto/Cipher/_raw_aesni.abi3.so
	Crypto/Cipher/_raw_arc2.abi3.so
	Crypto/Cipher/_raw_blowfish.abi3.so
	Crypto/Cipher/_raw_ocb.abi3.so
	Crypto/Cipher/_raw_ofb.abi3.so
	Crypto/Cipher/_ARC4.abi3.so
	Crypto/Cipher/_Salsa20.abi3.so
	Crypto/Cipher/_chacha20.abi3.so
	Crypto/Cipher/_raw_cast.abi3.so
	Crypto/Cipher/_raw_cbc.abi3.so
	Crypto/Cipher/_raw_cfb.abi3.so
	Crypto/Cipher/_raw_ctr.abi3.so
	Crypto/Cipher/_raw_des.abi3.so
	Crypto/Cipher/_raw_des3.abi3.so
	Crypto/Cipher/_raw_ecb.abi3.so
	Crypto/Cipher/_raw_eksblowfish.abi3.so
	Crypto/Hash/_BLAKE2s.abi3.so
	Crypto/Hash/_BLAKE2b.abi3.so
	Crypto/Hash/_SHA384.abi3.so
	Crypto/Hash/_SHA512.abi3.so
	Crypto/Hash/_ghash_clmul.abi3.so
	Crypto/Hash/_ghash_portable.abi3.so
	Crypto/Hash/_keccak.abi3.so
	Crypto/Hash/_poly1305.abi3.so
	Crypto/Hash/_MD2.abi3.so
	Crypto/Hash/_MD4.abi3.so
	Crypto/Hash/_MD5.abi3.so
	Crypto/Hash/_RIPEMD160.abi3.so
	Crypto/Hash/_SHA1.abi3.so
	Crypto/Hash/_SHA224.abi3.so
	Crypto/Hash/_SHA256.abi3.so

	Crypto/Math/_modexp.abi3.so
	Crypto/Protocol/_scrypt.abi3.so
	Crypto/PublicKey/_curve25519.abi3.so
	Crypto/PublicKey/_curve448.abi3.so
	Crypto/PublicKey/_ec_ws.abi3.so
	Crypto/PublicKey/_ed25519.abi3.so
	Crypto/PublicKey/_ed448.abi3.so
	Crypto/Util/_cpuid_c.abi3.so
	Crypto/Util/_strxor.abi3.so
	_cffi_backend.cpython-310-x86_64-linux-gnu.so
	google/protobuf/internal/_api_implementation.cpython-310-x86_64-linux-gnu.so
	google/protobuf/pyext/_message.cpython-310-x86_64-linux-gnu.so
	_ldap.cpython-310-x86_64-linux-gnu.so
	greenlet/_greenlet.cpython-310-x86_64-linux-gnu.so
	greenlet/tests/_test_extension.cpython-310-x86_64-linux-gnu.so
	greenlet/tests/_test_extension_cpp.cpython-310-x86_64-linux-gnu.so
	bcrypt/_bcrypt.abi3.so
	cb/response/proto/_c_ext.cpython-310-x86_64-linux-gnu.so
	cffi.libs/libffi-9c61262e.so.8.1.0
	coverage/tracer.cpython-310-x86_64-linux-gnu.so
	cryptography/hazmat/bindings/_rust.abi3.so
	markupsafe/_speedups.cpython-310-x86_64-linux-gnu.so
	gevent/_gevent_c_abstract_linkable.cpython-310-x86_64-linux-gnu.so
	gevent/_gevent_c_greenlet_primitives.cpython-310-x86_64-linux-gnu.so
	gevent/_gevent_c_hub_local.cpython-310-x86_64-linux-gnu.so
	gevent/libev/_corecffi.abi3.so
	gevent/libev/corecext.cpython-310-x86_64-linux-gnu.so
	gevent/_gevent_c_hub_primitives.cpython-310-x86_64-linux-gnu.so
	gevent/_gevent_c_ident.cpython-310-x86_64-linux-gnu.so
	gevent/_gevent_c_imap.cpython-310-x86_64-linux-gnu.so
	gevent/_gevent_c_semaphore.cpython-310-x86_64-linux-gnu.so
	gevent/_gevent_c_tracer.cpython-310-x86_64-linux-gnu.so
	gevent/libuv/_corecffi.abi3.so
	gevent/_gevent_c_waiter.cpython-310-x86_64-linux-gnu.so
	gevent/resolver/cares.cpython-310-x86_64-linux-gnu.so
	gevent/_gevent_cevent.cpython-310-x86_64-linux-gnu.so
	gevent/_gevent_cgreenlet.cpython-310-x86_64-linux-gnu.so

	gevent/_gevent_clocal.cpython-310-x86_64-linux-gnu.so
	gevent/_gevent_cqueue.cpython-310-x86_64-linux-gnu.so
	hiredis/hiredis.cpython-310-x86_64-linux-gnu.so
	msgspec/_core.cpython-310-x86_64-linux-gnu.so
	netifaces.cpython-310-x86_64-linux-gnu.so
	numpy/core/_multiarray_tests.cpython-310-x86_64-linux-gnu.so
	numpy/core/_multiarray_umath.cpython-310-x86_64-linux-gnu.so
	numpy/core/_operand_flag_tests.cpython-310-x86_64-linux-gnu.so
	numpy/core/_rational_tests.cpython-310-x86_64-linux-gnu.so
	numpy/core/_simd.cpython-310-x86_64-linux-gnu.so
	numpy/core/_struct_ufunc_tests.cpython-310-x86_64-linux-gnu.so
	numpy/core/_umath_tests.cpython-310-x86_64-linux-gnu.so
	numpy/fft/_pocketfft_internal.cpython-310-x86_64-linux-gnu.so
	numpy/linalg/_umath_linalg.cpython-310-x86_64-linux-gnu.so
	numpy/linalg/lapack_lite.cpython-310-x86_64-linux-gnu.so
	numpy/random/_bounded_integers.cpython-310-x86_64-linux-gnu.so
	numpy/random/_common.cpython-310-x86_64-linux-gnu.so
	numpy/random/_generator.cpython-310-x86_64-linux-gnu.so
	numpy/random/_mt19937.cpython-310-x86_64-linux-gnu.so
	numpy/random/_pcg64.cpython-310-x86_64-linux-gnu.so
	numpy/random/_philox.cpython-310-x86_64-linux-gnu.so
	numpy/random/_sfc64.cpython-310-x86_64-linux-gnu.so
	numpy/random/bit_generator.cpython-310-x86_64-linux-gnu.so
	numpy/random/mtrand.cpython-310-x86_64-linux-gnu.so
	numpy.libs/libgfortran-040039e1.so.5.0.0
	numpy.libs/libopenblas64_p-r0-2f7c42d4.3.18.so
	numpy.libs/libquadmath-96973f99.so.0.0.0
	zope/interface/_zope_interface_coptimizations.cpython-310-x86_64-linux-gnu.so
	psutil/_psutil_linux.cpython-310-x86_64-linux-gnu.so
	psutil/_psutil_posix.cpython-310-x86_64-linux-gnu.so
	psycopg2/_psycopg.cpython-310-x86_64-linux-gnu.so
	pvectorc.cpython-310-x86_64-linux-gnu.so
	rapidjson.cpython-310-x86_64-linux-gnu.so
	regex/_regex.cpython-310-x86_64-linux-gnu.so
	simplejson/_speedups.cpython-310-x86_64-linux-gnu.so
	sqlalchemy/cprocessors.cpython-310-x86_64-linux-gnu.so
	sqlalchemy/cresultproxy.cpython-310-x86_64-linux-gnu.so
	sqlalchemy/cutils.cpython-310-x86_64-linux-gnu.so

	wrapt/_wrappers.cpython-310-x86_64-linux-gnu.so
	selinux/_selinux.cpython-310-x86_64-linux-gnu.so
	selinux/audit2why.cpython-310-x86_64-linux-gnu.so
Xmlsec /usr/share/cb/lib/	libxmlsec1-openssl.so
OpenResty /usr/local/openresty/	zlib/lib/libz.so
	zlib/lib/libz.so.1
	zlib/lib/libz.so.1.2.12
	pcre/lib/libpcre.so
	pcre/lib/libpcre.so.1
	pcre/lib/libpcre.so.1.2.12
	luajit/lib/libluajit-5.1.so
	luajit/lib/libluajit-5.1.so.2
	luajit/lib/libluajit-5.1.so.2.1.0
	lualib/cjson.so
	lualib/librestysignal.so
	lualib/redis/parser.so
	nginx/modules/ngx_http_security_headers_module.so
pgsql-13 /usr/pgsql-13/lib/	libecpg.so
	fuzzystrmatch.so
	libecpg.so.6
	hstore.so
	libecpg.so.6.13
	hstore_plperl.so
	libecpg_compat.so.3
	cube.so
	libecpg_compat.so.3.13
	insert_username.so
	pg_freespacemap.so
	libpgtypes.so.3
	pg_prewarm.so
	libpgtypes.so.3.13
	libpq.so.5
	pg_stat_statements.so
	libpq.so.5.13
	pgcrypto.so
	libpqwalreceiver.so
	pg_visibility.so
	cyrillic_and_mic.so
	dict_int.so

pgrowlocks.so
dict_snowball.so
pgstattuple.so
dict_xsyn.so
pgxml.so
euc2004_sjis2004.so
postgres_fdw.so
euc_cn_and_mic.so
refint.so
euc_jp_and_sjis.so
seg.so
euc_kr_and_mic.so
sepgsql.so
euc_tw_and_big5.so
dblink.so
latin2_and_win1250.so
sslinfo.so
latin_and_mic.so
pgoutput.so
plpgsql.so
tablefunc.so
utf8_and_big5.so
pg_buffercache.so
utf8_and_cyrillic.so
tcn.so
utf8_and_euc2004.so
test_decoding.so
utf8_and_euc_cn.so
tsm_system_rows.so
utf8_and_euc_jp.so
tsm_system_time.so
utf8_and_euc_kr.so
unaccent.so
utf8_and_euc_tw.so
uuid-ossp.so
utf8_and_gb18030.so
utf8_and_gbk.so
utf8_and_iso8859.so
earthdistance.so
utf8_and_iso8859_1.so

	utf8_and_johab.so
	utf8_and_sjis.so
	file_fdw.so
	utf8_and_sjis2004.so
	utf8_and_uhc.so
	utf8_and_win.so
	_int.so
	adminpack.so
	amcheck.so
	auth_delay.so
	auto_explain.so
	autoinc.so
	bloom.so
	btree_gin.so
	btree_gist.so
	citext.so
	isn.so
	jsonb_plperl.so
	lo.so
	ltree.so
	moddatetime.so
	pageinspect.so
	passwordcheck.so
	pg_trgm.so

Additionally, see Appendix A and B for the listings of third-party library packages used when compiling the Server and user interface code as defined in the SBOM committed to NIAP. This list is in addition to the above dynamic libraries identified for this SFR.

[Host Sensor] The Host Sensor does not install any dynamic libraries on the platform.

Additionally, see Appendix C for a listing of third-party library packages used when compiling the Sensor code as defined in the SBOM committed to NIAP. This list is in addition to the above dynamic libraries identified for this SFR.

8.8.6 [App PP] FPT_TUD_EXT.1(Server), FPT_TUD_EXT.1(Sensor), and FPT_TUD_EXT.2

[EDR Server] The EDR Server application is an additional package to the OS platform. The product is packaged as an .rpm file format for the RHEL OS platform. The product's individual RPMs (installations and updates) are signed using CarbonBlack certificate via GPG signing. The YUM repo file is configured to validate GPG signatures automatically. YUM uses the CB public key, available from the same CB EDR REPO that the .rpm was downloaded from, to validate the signatures. With signature verification enabled, YUM will refuse to install any packages not GPG-signed with the correct key for that repository.

The binary code is only modified or replaced when a manual software update is performed from the OS. The TOE does not automatically download, modify, replace, or update its own binaries or executable files. The initial installation requires the Administrator to download a customer specific .rpm file from the CB website onto the host server platform and perform “yum install”.

The EDR Server and the platform OS provide the Administrator the ability to check the version of the EDR Server that is currently running on the machine. By using the Web GUI, the current version is displayed on the bottom right-hand corner of the management console on all webpages. The version of the EDR Server can also be verified by using the "yum info cb-enterprise" command.

The Administrator can check to see if an update is available by running the “yum check-update cb-enterprise” command. If there is an update available, the Administrator runs the “yum upgrade cb-enterprise” command that allows the EDR Server to be updated. The EDR Server does not download or install updates automatically.

Uninstallation of the EDR Server can be performed by the running the YUM utility to remove all remnants that constitute the EDR Server or remove packages plus all other third-party packages needed to operate. When the EDR Server application is uninstalled, settings are left behind for the purposes of re-installing the EDR Server software while maintaining previous configuration (recovery).

[Host Sensor] The Host Sensor application is an additional package to the OS platform. The product is packaged as an installer.exe and msi.exe formats for the Windows OS platform. The application software is digitally signed using a Carbon Black commercial CA certificate (authorized source) and then cross-signed by Microsoft’s WHQL signing. The software is verified by the EDR Server prior to being made available for installation on an Endpoint System. Unless the EDR Server has made an update package available for download, the Host Sensor does not automatically modify, replace, or update its own binaries or executable files. The Host Sensor supports functionality for both manual and automatic (default) updates.

If the Host Sensor has been configured for automatic updates:

> During each periodic polling cycle the Host Sensor application checks the EDR 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 Host Sensor has been configured for manual updates:

> The Local Administrator must navigate to the management server and authenticate as an Administrator and manually select the update package for downloading to the endpoint machine. The EDR 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 Local 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 Host Sensor will report that it is operating with updated version during the next polling cycle to the EDR Server. The EDR Server will then reset the upgrade request object to show that there is no update available.

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

The Host Sensor's OS platform provides multiple ways for the Local Administrator to check the version of the Host Sensor that is currently running on the machine.

- appwiz.cpl – once the programs and features windows display, scroll through list of applications until " 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.

8.8.6.1 *Timely Security Updates*

As part of providing timely security updates, Broadcom provides Carbon Black customers with a support portal, where they have the ability to submit support and security issues. This is an HTTPS site that requires user authentication prior to use. 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 contains third-party components that Carbon Black does not have control over the implementation of. Any implementation flaws are expected to be addressed within 90 days of reporting. Customers are notified of security advisories on the Broadcom customer support portal at https://support.broadcom.com/group/ecx/security-advisory.

8.9 Trusted Path/Channel

8.9.1 [App PP] FTP_DIT_EXT.1

[EDR Server] The EDR Server does not transfer data between itself and another trusted IT product in the operational environment.

[Host Sensor] The Host Sensor does not transfer data between itself and another trusted IT product in the operational environment.

8.9.2 [EDRMOD] FTP_TRP.1

The EDR Server invokes the underlying platform to establish a HTTPS (TLS v1.2) secure channel from the Remote Management Workstation to the TOE's Web GUI. The HTTPS connection enables remote administration of the TOE through the Remote Management Workstation. There is no other remote administrative interface used to manage the TOE.

9 Appendix A: EDR Server Third Party Libraries Obtained From SBOM

abego TreeLayout Core	ICU4J	Morfologik FSA
Aggregate Designer Algorithm	idna	Morfologik Stemming Dictionary for Polish
Annotations for Metrics	importlib_metadata	Morfologik stemming
ANTLR 4 Tool	io.grpc:grpc-alts	msgspec
antlr	io.grpc:grpc-auth	net.thisptr:jackson-jq
antlr4-python3-runtime	io.grpc:grpc-googleapis	netifaces
Aopalliance Version 1.0 Repackaged As A Module	io.grpc:grpc-grpclb	Netty Project
Apache Calcite Avatica Metrics	io.grpc:grpc-inprocess	Netty/Codec/Base
Apache Calcite Avatica	io.grpc:grpc-netty	Netty/Codec/Compression
Apache Commons BeanUtils	io.grpc:grpc-netty-shaded	Netty/TomcatNative [BoringSSL - Static]
Apache Commons Codec	io.grpc:grpc-protobuf-lite	Netty/TomcatNative [OpenSSL - Classes]
Apache Commons Collections	io.grpc:grpc-rls	Netty/Transport/Classes/Epoll
Apache Commons Compress	io.grpc:grpc-services	Netty/Transport/Classes/KQueue
Apache Commons Configuration	io.grpc:grpc-stub	Netty/Transport/Native/KQueue
Apache Commons Digester	io.grpc:grpc-util	Netty/Transport/Native/Unix/Common
Apache Commons Exec	io.grpc:grpc-xds	ntlm-auth
Apache Commons FileUpload	io.jaegertracing:jaeger-core	NumPy
Apache Commons IO	io.jaegertracing:jaeger-thrift	Oak Elasticsearch
Apache Commons Lang	io.swagger:swagger-annotations	OkHttp Server-Sent Events
Apache Commons Logging	istack common utility code runtime	OkHttp
Apache Commons Math	itsdangerous	OkIO
Apache Commons Pool	J2NET.Runtime.linux-x64	openai4j
Apache Commons Text	j2objc - Java to IOS	OpenCensus
Apache Commons Validator	Jackson 2 extensions to the Google HTTP Client Library for Java.	opencensus-api
Apache Hadoop Annotations	Jackson Integration for Metrics	opencsv
Apache Hadoop Auth	Jackson module: Jakarta XML Bind Annotations (jakarta.xml.bind)	OpenNLP Tools
Apache Hadoop Client API	jackson-annotations	OpenResty
Apache Hadoop Client Runtime	jackson-bom	OpenTelemetry Java - io.opentelemetry:opentelemetry-sdk-metrics
Apache Hadoop HDFS Client	jackson-core	OpenTelemetry Java
Apache Hadoop shaded Guava	jackson-databind	OpenTelemetry

Apache Hadoop	jackson-dataformat-binary	OpenTracing API
Apache HTTP transport v2 for the Google HTTP Client Library for Java.	jackson-dataformat-csv	OpenTracing-noop
Apache HttpClient	jackson-dataformat-smile	OpenTracing-util
Apache HTTPComponents Core	jackson-dataformat-yaml	org.apache.lucene.queryparser
Apache HttpMime	jackson-module-scala	org.apiguardian:apiguardian-api
Apache Kafka	jackson-modules-java8	org.conscrypt:conscrypt-openjdk-uber
Apache Log4J API	JaCoCo	org.jetbrains.kotlin:kotlin-daemon-embeddable
Apache Log4j Layout for JSON template	Jakarta Activation	org.jetbrains.kotlin:kotlin-klib-commonizer-embeddable
Apache Log4j SLF4J 2.0 Binding	Jakarta Annotations API	org.jetbrains.kotlin:kotlin-scripting-common
Apache Log4j SLF4J Binding	Jakarta Dependency Injection	org.jetbrains.kotlin:kotlin-scripting-compiler-embeddable
Apache Log4j Web Adapters	jakarta.xml.bind:jakarta.xml.bind-api	org.jetbrains.kotlin:kotlin-scripting-compiler-impl-embeddable
Apache Log4j	jakartaee/rest	org.jetbrains.kotlin:kotlin-scripting-jvm
Apache Lucene (module: common)	jakartaeevalidation	org.jetbrains.kotlin:kotlin-stdlib-jdk7
Apache Lucene (module: icu)	Janino	org.jetbrains.kotlin:kotlin-stdlib-jdk8
Apache Lucene (module: kuromoji)	jansi	org.jetbrains.kotlin:kotlin-test-annotations-common
Apache Lucene (module: morfologik)	jaraco/jaraco.functools	org.jetbrains.kotlin:kotlin-test-junit5
Apache Lucene (module: nori)	jaraco/jaraco.text	org.junit.jupiter:junit-jupiter-api
Apache Lucene (module: opennlp)	Java API for Processing JSON (JSON-P)	org.junit.jupiter:junit-jupiter-engine
Apache Lucene (module: phonetic)	Java Diff Utils	org.junit.jupiter:junit-jupiter-params
Apache Lucene (module: smartcn)	Java Native Access (JNA)	org.junit.platform:junit-platform-commons
Apache Lucene (module: stempel)	Java Servlet API	org.junit.platform:junit-platform-engine
Apache Lucene	Javassist	org.junit.platform:junit-platform-launcher
Apache POI - OOXML Schemas (lite)	javax.transaction API	org.junit.platform:junit-platform-suite-api
Apache POI	jaxb-api	org.junit.vintage:junit-vintage-engine
Apache Solr (module: api)	JCL 1.2 Implemented Over SLF4J	org.locationtech.jts.io:jts-io-common
Apache Solr (module: cross-dc)	jedis	org.locationtech.jts:jts-core
Apache Solr (module: cross-dc-manager)	jemalloc	org.opentest4j:opentest4j
Apache Solr (module: hadoop-auth)	Jersey Inject HK2	OSGi resource locator bundle
Apache Solr (module: hdfs)	Jersey	Packaging
Apache Solr (module: jwt-auth)	jersey-container-jetty-http	ParaNamer Core

Apache Solr (module: llm)	jersey-core-server	PathJson
Apache Solr (module: opentelemetry)	jersey-ext-entity-filtering	pbr
Apache Solr (module: scripting)	jersey-media-json-jackson	PCollections
Apache Solr (module: solrj-streaming)	Jetbrains annotations	PCRE
Apache Solr (module: solrj-zookeeper)	Jetty :: ALPN :: API	perfmark:perfmark-api
Apache Solr (module: sql)	Jetty :: ALPN :: JDK9 Client Implementation	pg8000
Apache Solr Analysis Extras	Jetty :: ALPN :: JDK9 Server Implementation	pid
Apache Solr Analytics Package	Jetty :: ALPN :: OpenJDK8 Client Implementation	pip
Apache Solr GCS BackupRepository	Jetty :: ALPN Client	platformdirs
Apache Solr Jaeger Tracer Configurator Package	Jetty :: ALPN Server	PostgreSQL Database Server
Apache Solr Language Identifier	Jetty :: HTTP2 :: Client	PostgreSQL JDBC Driver (pgjdbc)
Apache Solr Prometheus Exporter Package	Jetty :: HTTP2 :: Common	progressbar
Apache Solr S3 BackupRepository	Jetty :: HTTP2 :: HPACK	Proj4J
Apache Solr Solrj	Jetty :: HTTP2 :: HTTP Client Transport	Project Lombok
Apache Solr	Jetty :: Servlet API and Schemas for JPMS and OSGi	Prometheus Java Simpleclient Common
Apache Thrift	Jetty: Java based HTTP/1.x HTTP/2 Servlet WebSocket Server	Prometheus Java Simpleclient Httpserver
Apache Tika	jinjapython	Prometheus Java Simpleclient
Apache Xerces2 J	JLine - Java Console input Library	Prometheus Metrics Exposition Formats
Apache Yetus - Audience Annotations	JLine JANSI Terminal	Prometheus Metrics Model
Apache ZooKeeper - Jute	JLine Terminal	protobuf3-to-dict
Apache ZooKeeper	JMES Path Query library	protobuf-java
Arcade Data	jmespath.py	Protocol Buffer Java Util Package
argparse4j	Joda Time	Protocol Buffers [Core]
ASM based accessors helper used by json-smart	jolokia-core	proto-google-cloud-storage-v2
ASM Commons	jolokia-jvm	psf-requests
ASM Tree	jolokia-war-unsecured	psutil
ASM	JOpt Simple	psycogreen
asm-analysis	jose4j	psycopg2
asn1crypto	jsonic	Public Domain
atomicfu	json-simple	py
AutoValue Annotations	json-smart	py3-autocommand
AWS Event Stream	jsr107	py3-inflect
AWS Java SDK :: Arns	JTokkit	py3-jaraco.context
AWS Java SDK :: AWS CRT Core	JUL to SLF4J bridge	py3-tomli
AWS Java SDK :: Checksums SPI	JUnit 5 (Bill of Materials)	py-amqp

AWS Java SDK :: Checksums	JUnit Jupiter (Aggregator)	pyasn1
AWS Java SDK :: Core :: Protocols :: AWS Xml Protocol	JUnit	pyasn1-modules
AWS Java SDK :: Core :: Protocols :: Json Utils	JVM Integration for Metrics	pyca/cryptography
AWS Java SDK :: Endpoints SPI	Kerby ASN1 Project	pycparser
AWS Java SDK :: HTTP Auth AWS	Kerby Config	pycryptodome
AWS Java SDK :: HTTP Auth Event Stream	Kerby PKIX Project	pyformance
AWS Java SDK :: HTTP Auth SPI	Kerby Util	pyOpenSSL
AWS Java SDK :: HTTP Auth	Kerby-kerb core	pypi/setuptools
AWS Java SDK :: Identity SPI	Kerby-kerb Crypto	pypi-zope.interface
AWS Java SDK :: Metrics Interface	Kerby-kerb Util	pyrabbit2
AWS Java SDK :: Retries API	koin-core	pyrsistent
AWS Java SDK :: Retries	koin-ktor	pysaml2
AWS Java SDK :: Services :: Amazon S3	koin-logger-slf4j	pytest
AWS SDK for Java	koin-test	pytest-forked
bcrypt	kombu	pytest-html
BeautifulSoup4	Kotest	pytest-metadata
boto3	Kotlin	pytest-xdist
botocore	kotlin-compiler	python cffi
Bouncy Castle ASN.1 Extension and Utility APIs	kotlin-logging	Python Faker
Bouncy Castle PKIX CMS EAC TSP PKCS OCSP CMP and CRMF APIs	kotlin-reflect	Python programming language
Bouncy Castle	kotlin-script-runtime	Python six
Byte Buddy	kotlin-stdlib-common	Python Software Foundation License 2.3
byte-buddy-agent	kotlin-test	Python tzdata
cabalmock	kotlin-test-common	python3-charset-normalizer
cachelib	kotlinx-coroutines-bom	python-attrs
Caffeine cache	kotlinx-coroutines-core	python-autocommand
Calcite Core	kotlinx-coroutines-debug	python-certifi
Calcite Linq4j	kotlinx-coroutines-jdk8	python-dateutil
Carrot2 Core (algorithms and data sources)	kotlinx-coroutines-slf4j	python-decorator
ch.randelshofer:fastdoubleparser	kotlinx-coroutines-test	python-distlib
Checker Qual	ktor-client-cio	python-jsonschema
ClassGraph	ktor-client-core	python-ldap
Click - Python Command Line Utility	ktor-client-core-jvm	python-magic
Clikt	ktor-client-jetty	Python-Markdown
Cloud Storage API v1beta1 (revision 28)	ktor-events	python-netaddr
colorama	ktor-http	python-pluggy
colormath	ktor-http-cio	python-protobuf

com.google.api.grpc:proto-google-common-protos	ktor-http-jvm	python-rapidjson
com.google.api.grpc:proto-google-iam-v1	ktor-io	python-typeguard
com.helger:profiler	ktor-network	python-webencodings
com.yahoo.datasketches:memory	ktor-network-tls	python-wheel
com.yahoo.datasketches:sketches-core	ktor-network-tls-certificates	PyTZ - Python Time Zone Library
commons-cli	ktor-serialization	pytz-deprecation-shim
commons-dbcp	ktor-server	RabbitMQ amqp-client
config - com.typesafe:config	ktor-server-auto-head-response	rabbitmq/erlang-rpm
Converter: Jackson	ktor-server-caching-headers	rabbitmq-server
CopernicaMarketingSoftware/AMQP-CPP	ktor-server-call-id	re2j
Coverage	ktor-server-call-logging	reactive-streams-jvm
cowboy server	ktor-server-compression	redis
Curator Client	ktor-server-conditional-headers	redis-py
Curator Framework	ktor-server-content-negotiation	regex
Curator Recipes	ktor-server-core	Repoze Public License
curvesapi	ktor-server-cors	requests-mock
dateparser	ktor-server-data-conversion	RgxGen
defusedxml	ktor-server-default-headers	RocksDB JNI
Deprecated	ktor-server-double-receive	RonnyPfannschmidt/iniconfig
Disruptor Framework	ktor-server-forwarded-header	RustPython
disruptor	ktor-server-host-common	s3transfer
duo_client_python	ktor-server-hsts	Scala
EasyMock	ktor-server-http-redirect	scala-collection-compat
easymockobjenesis	ktor-server-method-override	scala-java8-compat
Eclipse Distribution (Temurin-17)	ktor-server-netty	scala-logging
Ehcache	ktor-server-partial-content	scramp
elementpath	ktor-server-sessions	semver4j
erlang-otp	ktor-server-status-pages	ServiceLocator Default Implementation
erlang-runtime_tools	ktor-server-test-host	Simple JMX
error-prone annotations	ktor-server-tests	simplejson
ESG-CBEDR-Server	ktor-server-websockets	SLF4J API Module
execnet	ktor-test-dispatcher	slf4j-nop
exposed-core	ktor-utils	SnakeYAML
exposed-dao	ktor-utils-jvm	snappy-java
exposed-java-time	ktor-websockets	solr-cmd-utils
exposed-jdbc	ktor-websocket-serialization	sortedcontainers
express-http-proxy	LangChain4j integration with Cohere	soupsieve
expressjs/express	LangChain4j integration with Hugging Face	SparseBitSet

fakeredis	LangChain4j integration with MistralAI	Spatial4j
FindBugs jsr305	LangChain4j integration with OpenAI	SQLAlchemy
Flask	langchain4j-core	sqlalchemy-migrate
Flask-SeaSurf	language-detection	sqlalchemy-views
Flask-Session	libpq	sqlparse
Flask-Testing	Log4J Compatibility API	square-retrofit
flot.github.com	Log4j Implemented Over SLF4J	Stately
gapic-google-cloud-storage-v2	Logback	stax2-api
GAX (Google Api eXtensions)	lua-nginx-module	stringtemplate4
gevent	Lucene Classification	Sun GPL With Classpath Exception v2.0
gevent-websocket	Lucene codecs	Supervisor
GNU General Public License v2.0 w/Classpath exception	Lucene Expressions	Swagger Parser
GNU Lesser General Public License v3.0 with Exceptions	Lucene Facets	swagger-annotations-jakarta
Go programming language	Lucene Grouping	swagger-ui
Google Android Annotations Library	Lucene Highlighter	tabulate
Google API Common	Lucene Join	t-digest
Google APIs Client Library for Java	Lucene Memory	teamcity-messages
Google App Engine extensions to the Google HTTP Client Library for Java.	Lucene Miscellaneous	Tempita
Google Auth Library for Java	Lucene Queries	termcolor-pypi
Google Cloud Core	Lucene Query Parser	testing.common.database
Google Cloud Storage	Lucene Sandbox	testing.postgresql
Google HTTP Client Library for Java	Lucene Spatial3D	testing.redis
Google S2 geometry library	Lucene Spatial Extras	text-unidecode
googleapis/google-cloud-java	Lucene Suggest	ThreeTen backport
googlegson	lucene-backward-codecs	tomli
googleguava	lz4java	Trove4J
google-oauth-java-client	MarkupSafe	typeguard
Graphite Integration for Metrics	Metrics Core Library	typing_extensions
greenlet	Metrics Core	tzlocal
grequests	Metrics Health Checks	Ukrainian POS tag dictionary for Morfologik
grpc-google-cloud-storage-v2	Metrics Integration for Jetty 10.x and higher	urllib3
grpc-java	Metrics Integration for Jetty 9.1 and higher	Uzaygezen-core
GSON extensions to the Google HTTP Client Library for Java.	Metrics Integration with JMX	vine
Guava InternalFutureFailureAccess and InternalFutures	Metrics Utility Servlets	Werkzeug
Guava ListenableFuture only	Mockito_	Woodstox
Gunicorn	MockK Agent API	wrapt

hadoop-mapreduce-client-common	MockK DSL common	XMLBeans
Hamcrest	MockK Java Agent API	xmlschema
hazelcast	MockK Java Agent	yargs
hiredis	MockK Java DSL	zipp
HK2 API module	MockK	zope.event
HK2 Implementation Utilities	mordant	zstd-jni
HPPC	more-itertools	

10 Appendix B: EDR Server User Interface Third Party Libraries Obtained From SBOM

@ampproject/remapping	d3-path	nodejs-walker
@babel/code-frame	d3-path	node-lru-cache
@babel/compat-data	d3-sankey	node-lru-cache
@babel/core	d3-scale	node-mkdirp
@babel/eslint-parser	d3-shape	node-process
@babel/generator	d3-shape	node-releases
@babel/helper-annotate-as-pure	d3-time	node-semver
@babel/helper-builder-binary-assignment-operator-visitor	d3-time-format	node-semver
@babel/helper-compilation-targets	d3-voronoi	node-semver
@babel/helper-create-class-features-plugin	damerau-levenshtein	node-shell-quote
@babel/helper-create-regexp-features-plugin	DamonOehlman/wildcard	node-source-map-support
@babel/helper-define-polyfill-provider	data-urls	node-tmp
@babel/helper-member-expression-to-functions	data-view-buffer	normalize-package-data
@babel/helper-module-imports	data-view-byte-length	normalize-package-data
@babel/helper-module-transforms	data-view-byte-offset	normalize-path
@babel/helper-optimise-call-expression	date-fns/date-fns	normalize-range
@babel/helper-plugin-utils	debug-js/debug	normalize-selector
@babel/helper-remap-async-to-generator	debug-js/debug	npm ini
@babel/helper-replace-supers	debug-js/debug	npm-run-path
@babel/helpers	debug-js/debug	nth-check
@babel/helper-skip-transparent-expression-wrappers	debug-js/debug	nwsapi
@babel/helper-string-parser	debug-js/debug	object.entries
@babel/helper-validator-identifier	debug-js/debug	object.fromentries
@babel/helper-validator-option	debug-js/debug	object.groupby
@babel/helper-wrap-function	Decamelize	object.hasown
@babel/parser	decamelize-keys	object.values
@babel/plugin-bugfix-firefox-class-in-computed-class-key	decimal.js	object-assign

@babel/plugin-bugfix-safari-class-field-initializer-scope	decode-uri-component	object-inspect
@babel/plugin-bugfix-safari-id-destructuring-collision-in-function-expression	dedent	object-is
@babel/plugin-bugfix-v8-spread-parameters-in-optional-chaining	deep-is	object-keys
@babel/plugin-bugfix-v8-static-class-fields-redefine-readonly	deepmerge	obuf
@babel/plugin-proposal-export-default-from	default-gateway	on-finished
@babel/plugin-proposal-private-property-in-object	define-data-property	on-headers
@babel/plugin-syntax-async-generators	define-lazy-prop	optionator
@babel/plugin-syntax-bigint	define-properties	os-tmpdir
@babel/plugin-syntax-class-properties	delayed-stream	own-keys
@babel/plugin-syntax-class-static-block	destroy	PapaParse
@babel/plugin-syntax-dynamic-import	detect-newline	param-case
@babel/plugin-syntax-export-namespace-from	detect-node	parent-module
@babel/plugin-syntax-import-assertions	dir-glob	parse5
@babel/plugin-syntax-import-attributes	dns-packet	parse-json
@babel/plugin-syntax-import-meta	doctrine	parse-key
@babel/plugin-syntax-json-strings	doctrine	parse-node-version
@babel/plugin-syntax-jsx	dom-accessibility-api	parse-srcset
@babel/plugin-syntax-logical-assignment-operators	dom-align	pascal-case
@babel/plugin-syntax-nullish-coalescing-operator	dom-converter	patch-package
@babel/plugin-syntax-numeric-separator	domelementtype	path-exists
@babel/plugin-syntax-object-rest-spread	domexception	path-is-absolute
@babel/plugin-syntax-optional-catch-binding	DomHandler	path-key
@babel/plugin-syntax-optional-chaining	DomHandler	path-key
@babel/plugin-syntax-private-property-in-object	dom-helpers	path-parse
@babel/plugin-syntax-top-level-await	dom-serializer	path-to-regexp
@babel/plugin-syntax-typescript	dom-serializer	path-to-regexp
@babel/plugin-syntax-unicode-sets-regex	domutils	path-type
@babel/plugin-transform-arrow-functions	domutils	peerigon/extract-loader
@babel/plugin-transform-async-generator-functions	domutils	performance-now
@babel/plugin-transform-async-to-generator	dom-walk	performance-now
@babel/plugin-transform-block-scoped-functions	dot-case	picocolors
@babel/plugin-transform-block-scoping	dunder-proto	picomatch
@babel/plugin-transform-classes	dylang/node-xml	pidtree
@babel/plugin-transform-class-properties	eastasianwidth	pify
@babel/plugin-transform-class-static-block	ee-first	pify
@babel/plugin-transform-computed-properties	einaros/ws	pillarjs/parseurl
@babel/plugin-transform-destructuring	einaros/ws	pinkie-promise
@babel/plugin-transform-dotall-regex	electron-to-chromium	pirates
@babel/plugin-transform-duplicate-keys	emaphp/underscore-template-loader	pkg-dir

@babel/plugin-transform-duplicate-named-capturing-groups-regex	emittery	p-limit
@babel/plugin-transform-dynamic-import	emoji-regex	p-limit
@babel/plugin-transform-exponentiation-operator	emoji-regex	p-locate
@babel/plugin-transform-export-namespace-from	emojis-list	p-locate
@babel/plugin-transform-for-of	encodeurl	p-map
@babel/plugin-transform-function-name	encodeurl	possible-typed-array-names
@babel/plugin-transform-json-strings	enhanced-resolve	PostCSS
@babel/plugin-transform-literals	enhanced-resolve	postcss-attribute-case-insensitive
@babel/plugin-transform-logical-assignment-operators	envinfo	postcss-calc
@babel/plugin-transform-member-expression-literals	errno	postcss-clamp
@babel/plugin-transform-modules-amd	error-stack-parser	postcss-color-functional-notation
@babel/plugin-transform-modules-commonjs	es-abstract	postcss-color-hex-alpha
@babel/plugin-transform-modules-systemjs	escalade	postcss-colormin
@babel/plugin-transform-modules-umd	escape-html	postcss-color-rebeccapurple
@babel/plugin-transform-named-capturing-groups-regex	escape-string-regexp	postcss-convert-values
@babel/plugin-transform-new-target	escape-string-regexp	postcss-custom-media
@babel/plugin-transform-nullish-coalescing-operator	escape-string-regexp	postcss-custom-properties
@babel/plugin-transform-numeric-separator	escodegen	postcss-custom-selectors
@babel/plugin-transform-object-rest-spread	es-define-property	postcss-dir-pseudo-class
@babel/plugin-transform-object-super	es-errors	postcss-discard-comments
@babel/plugin-transform-optional-catch-binding	ESG-CBEDR-Server	postcss-discard-duplicates
@babel/plugin-transform-optional-chaining	ESLint	postcss-discard-empty
@babel/plugin-transform-parameters	eslint-config-prettier	postcss-discard-overridden
@babel/plugin-transform-private-methods	eslint-import-resolver-node	postcss-double-position-gradients
@babel/plugin-transform-private-property-in-object	eslint-plugin-import	postcss-env-function
@babel/plugin-transform-property-literals	eslint-plugin-import	postcss-focus-visible
@babel/plugin-transform-react-constant-elements	eslint-plugin-import	postcss-focus-within
@babel/plugin-transform-react-display-name	eslint-plugin-jsx-a11y	postcss-font-variant
@babel/plugin-transform-react-jsx	eslint-plugin-prettier	postcss-gap-properties
@babel/plugin-transform-react-jsx-development	eslint-plugin-react-hooks	postcss-image-set-function
@babel/plugin-transform-react-pure-annotations	eslint-plugin-simple-import-sort	postcss-initial
@babel/plugin-transform-regenerator	eslint-scope	postcss-lab-function
@babel/plugin-transform-reserved-words	eslint-scope	postcss-less
@babel/plugin-transform-runtime	eslint-visitor-keys	postcss-loader
@babel/plugin-transform-shorthand-properties	eslint-visitor-keys	postcss-logical
@babel/plugin-transform-spread	es-module-lexer	postcss-media-minmax
@babel/plugin-transform-sticky-regex	es-object-atoms	postcss-media-query-parser

@babel/plugin-transform-template-literals	espree	postcss-merge-longhand
@babel/plugin-transform-typeof-symbol	Esprima	postcss-merge-rules
@babel/plugin-transform-typescript	esquery	postcss-minify-font-values
@babel/plugin-transform-unicode-escapes	esrecurse	postcss-minify-gradients
@babel/plugin-transform-unicode-property-regex	es-set-tostringtag	postcss-minify-params
@babel/plugin-transform-unicode-regex	es-shim-unscopables	postcss-minify-selectors
@babel/plugin-transform-unicode-sets-regex	es-to-primitive	postcss-modules-extract-imports
@babel/preset-env	estraverse	postcss-modules-local-by-default
@babel/preset-modules	estraverse	postcss-modules-scope
@babel/preset-react	estraverse	postcss-modules-values
@babel/preset-typescript	esutils	postcss-nesting
@babel/regjsgen	etag	postcss-normalize-charset
@babel/runtime	eventemitter3	postcss-normalize-display-values
@babel/runtime-corejs3	execa	postcss-normalize-positions
@babel/template	execall	postcss-normalize-repeat-style
@babel/types	expect	postcss-normalize-string
@bcoe/v8-coverage	exports-loader	postcss-normalize-timing-functions
@csstools/postcss-font-format-keywords	expressjs/accepts	postcss-normalize-unicode
@csstools/postcss-hwb-function	expressjs/compressible	postcss-normalize-url
@csstools/postcss-is-pseudo-class	expressjs/express	postcss-normalize-whitespace
@csstools/postcss-normalize-display-values	expressjs/mime-types	postcss-opacity-percentage
@csstools/postcss-progressive-custom-properties	fast-deep-equal	postcss-ordered-values
@discoveryjs/json-ext	fast-diff	postcss-overflow-shorthand
@emotion/babel-plugin	fastest-levenshtein	postcss-page-break
@emotion/cache	fast-glob	postcss-place
@emotion/hash	fast-json-stable-stringify	postcss-preset-env
@emotion/is-prop-valid	fast-levenshtein	postcss-pseudo-class-any-link
@emotion/memoize	fastparse	postcss-reduce-initial
@emotion/react	fastq	postcss-reduce-transforms
@emotion/serialize	fast-uri	postcss-replace-overflow-wrap
@emotion/sheet	faye-websocket-node	postcss-resolve-nested-selector
@emotion/styled	felixge/node-retry	postcss-safe-parser
@emotion/unitless	file-entry-cache	postcss-selector-not
@emotion/use-insertion-effect-with-fallbacks	file-loader	postcss-selector-parser
@emotion/utils	FileSaver.js	postcss-svgo
@emotion/weak-memoize	fill-range	postcss-unique-selectors
@emotion/weak-memoize	finalhandler	postcss-value-parser
@eslint/eslintrc	find-cache-dir	prelude-ls
@eslint/js	find-root	p-retry
@eslint-community/eslint-utils	find-up	Prettier IO

@eslint-community/regexpp	find-up	prettier-linter-helpers
@fortawesome/fontawesome-free	find-yarn-workspace-root	pretty-error
@humanwhocodes/config-array	flat-cache	pretty-format
@humanwhocodes/module-importer	flatted	pretty-format
@humanwhocodes/object-schema	floatdrop/pinkie	process-nextick-args
@istanbuljs/load-nyc-config	follow-redirects	prompts
@istanbuljs/schema	form-data	prop-types
@jest/console	form-data	proxy-addr
@jest/core	forwarded	proxy-from-env
@jest/environment	fraction.js	prr
@jest/fake-timers	fs.realpath	psl
@jest/globals	fsevents	p-try
@jest/reporters	fs-monkey	Punycode.js
@jest/source-map	function.prototype.name	Qix/color
@jest/test-result	function-bind	Qix-/color-convert
@jest/test-sequencer	functions-have-names	Qix-/color-convert
@jest/transform	gajus/table	qs - QS Querystring
@jest/types	gensync	query-string
@jest/types	get-caller-file	querystringify
@jridgewell/gen-mapping	get-intrinsic	queue-microtask
@jridgewell/gen-mapping	get-package-type	quick-lru
@jridgewell/resolve-uri	get-proto	rackt/react-router-redux
@jridgewell/set-array	get-stdin	randombytes
@jridgewell/source-map	get-stream	range-parser
@jridgewell/sourcemap-codec	get-symbol-description	raw-body
@jridgewell/trace-mapping	global	Raynos/for-each
@leichtgewicht/ip-codec	global-modules	rc-align
@mui/base	global-prefix	rc-animate
@mui/core-downloads-tracker	globalthis	rc-tooltip
@mui/icons-material	globjoin	rc-trigger
@mui/material	glob-parent	rc-util
@mui/private-theming	glob-parent	React Date-Range
@mui/styled-engine	glob-parent	React from Facebook
@mui/system	glob-to-regexp	react-addons-shallow-compare
@mui/types	gopd	react-base16-styling
@mui/utils	Gozala/events	react-dock
@nicolo-ribaudo/eslint-scope-5-internals	graphemer	react-dom
@nodelib/fs.scandir	handle-thing	react-draggable
@nodelib/fs.stat	hard-rejection	react-grid-layout
@nodelib/fs.walk	harmony-reflect	react-i18next
@pmmmwh/react-refresh-webpack-plugin	has	react-immutable-proptypes
@popperjs/core	has-ansi	react-is

@primer/octicons-react	has-bigints	react-is
@redux-devtools/core	has-flag	react-is
@redux-devtools/dock-monitor	hasown	react-is
@redux-devtools/instrument	has-property-descriptors	react-is
@redux-devtools/log-monitor	has-proto	ReactiveX RxJS
@rtsao/scc	has-symbols	ReactiveX RxJS
@sinonjs/commons	has-tostringtag	react-json-tree
@sinonjs/fake-timers	hex-color-regex	react-lifecycles-compat
@svgr/babel-plugin-add-jsx-attribute	History	react-list
@svgr/babel-plugin-remove-jsx-attribute	hoist-non-react-statics	react-motion
@svgr/babel-plugin-remove-jsx-empty-expression	hoist-non-react-statics	react-onclickoutside
@svgr/babel-plugin-replace-jsx-attribute-value	hosted-git-info	reactredux
@svgr/babel-plugin-svg-dynamic-title	hosted-git-info	react-refresh
@svgr/babel-plugin-svg-em-dimensions	hpack.js	react-resizable
@svgr/babel-plugin-transform-react-native-svg	hsla-regex	react-router-dom
@svgr/babel-plugin-transform-svg-component	hsl-regex	reacttraining/react-router
@svgr/babel-preset	html-encoding-sniffer	react-transition-group
@svgr/core	html-loader-entozoon	react-vis
@svgr/hast-util-to-babel-ast	html-minifier-terser	readable-stream
@svgr/plugin-jsx	htmlparser2	readable-stream
@svgr/plugin-svgo	htmlparser2	readdirp
@svgr/webpack	html-parse-stringify	read-pkg
@testing-library/dom	html-tags	read-pkg-up
@testing-library/jest-dom	html-webpack-plugin	rechoir
@testing-library/react	http-deceiver	redent
@tootallnate/once	http-errors	Redux JS
@trysound/sax	http-errors	Redux JS
@trysound/sax	http-errors	redux-devtools-themes
@types/aria-query	http-parser-js	redux-mock-store
@types/babel__core	http-proxy	redux-thunk
@types/babel__generator	http-proxy-middleware	reflect.getprototypeof
@types/babel__template	https-proxy-agent	regenerate-unicode-properties
@types/babel__traverse	human-signals	regenerator-runtime
@types/base16	i18next	regenerator-runtime
@types/body-parser	i18next-browser-languagedetector	regenerator-transform
@types/bonjour	i18next-xhr-backend	regexp.prototype.flags
@types/connect	iarna/write-file-atomic	regexpu-core
@types/connect-history-api-fallback	iarna/write-file-atomic	regjsparser
@types/estree	iconv-lite	relateurl
@types/express	iconv-lite	renderkid
@types/express-serve-static-core	icss-utils	require-directory
@types/graceful-fs	immutable-js	require-from-string

@types/hoist-non-react-statics	import-fresh	requires-port
@types/html-minifier-terser	import-lazy	reselect
@types/http-proxy	import-local	resolve.exports
@types/istanbul-lib-coverage	imports-loader	resolve-cwd
@types/istanbul-lib-report	imurmurhash	resolve-from
@types/istanbul-reports	indent-string	resolve-from
@types/jest	indexof-component	resolve-pathname
@types/json5	inflight	restore-cursor
@types/json-schema	inherits	reusify
@types/lodash	inherits	rfdc
@types/lodash.debounce	internal-slot	rgba-regex
@types/marked	internmap	rgb-regex
@types/mime	internmap	rich-harris/svg-parser
@types/minimist	interpret	rimraf
@types/node	interpret	rimraf
@types/node-forge	invariant	rpm-builder
@types/normalize-package-data	invariant	run-parallel
@types/papaparse	isaacs/once	safe-array-concat
@types/parse-json	is-absolute-url	safe-buffer
@types/prettier	is-arguments	safe-buffer
@types/prop-types	isarray	safe-push-apply
@types/qs	isarray	safer-buffer
@types/range-parser	isarray	safe-regex-test
@types/react	is-array-buffer	samccone/chrome-trace-event
@types/react-redux	is-arrayish	sanitize-html
@types/react-transition-group	is-arrayish	saxes
@types/redux-devtools-themes	is-async-function	sax-js
@types/retry	is-bigint	scheduler
@types/scheduler	is-binary-path	schema-utils
@types/serve-index	is-boolean-object	schema-utils
@types/serve-static	is-callable	schema-utils
@types/sockjs	is-ci	schema-utils
@types/stack-utils	is-color-stop	select-hose
@types/testing-library__jest-dom	is-core-module	selfsigned
@types/ws	is-data-view	send
@types/yargs	is-date-object	serialize-javascript
@types/yargs	is-docker	serve-index
@types/yargs-parser	isexe	serve-static
@ungap/structured-clone	is-extglob	set-function-length
@webassemblyjs/ast	is-finalizationregistry	set-function-name
@webassemblyjs/floating-point-hex-parser	is-fullwidth-code-point	set-proto
@webassemblyjs/helper-api-error	is-fullwidth-code-point	setprototypeof
@webassemblyjs/helper-buffer	is-fullwidth-code-point	setprototypeof

@webassemblyjs/helper-numbers	is-generator-fn	shallow-clone
@webassemblyjs/helper-wasm-bytecode	is-generator-function	shallowequal
@webassemblyjs/helper-wasm-section	is-glob	shallow-equal
@webassemblyjs/ieee754	is-map	shebang-command
@webassemblyjs/leb128	is-negative-zero	shebang-command
@webassemblyjs/utf8	is-number	shebang-regex
@webassemblyjs/wasm-edit	is-number-object	shebang-regex
@webassemblyjs/wasm-gen	isobject	shortid
@webassemblyjs/wasm-opt	is-path-inside	side-channel
@webassemblyjs/wasm-parser	is-plain-obj	side-channel-list
@webassemblyjs/wast-printer	is-plain-obj	side-channel-map
@webpack-cli/configtest	is-plain-object	side-channel-weakmap
@webpack-cli/info	is-plain-object	simple-swizzle
@webpack-cli/serve	is-potential-custom-element-name	sindresorhus/globals
@xtuc/ieee754	is-regex	sindresorhus/globals
@xtuc/long	is-regexp	sindresorhus/globby
@yarnpkg/lockfile	is-resolvable	sindresorhus/globby
abab	is-set	sindresorhus/globby
Acorn	is-shared-array-buffer	sindresorhus/normalize-url
Acorn	is-stream	sindresorhus/onetime
acorn-globals	is-string	sindresorhus/slash
acorn-import-attributes	is-symbol	sindresorhus/slash
Acorn-JSX	istanbul-lib-coverage	sindresorhus/slash
acorn-walk	istanbul-lib-instrument	sindresorhus-open
add-dom-event-listener	istanbul-lib-report	sindresorhus-open
agent-base	istanbul-lib-source-maps	sisteransi
aggregate-error	istanbul-reports	slice-ansi
airbnb javascript	is-typed-array	slice-ansi
airbnb javascript	is-typed-array	slice-ansi
ajv	is-weakmap	snake-case
ajv	is-weakref	sockjs-node
ajv	is-weakset	source-map
ajv	is-what	source-map
ajv-formats	is-wsl	source-map
ajv-keywords	Jest from Facebook	source-map-js
ajv-keywords	Jest from Facebook	source-map-loader
alphanum-sort	jest-changed-files	source-map-resolve
ansi-escapes	jest-config	spawn-command
ansi-html	jest-diff	spdx-correct.js
ansi-html-community	jest-diff	spdx-expression-parse.js
ansi-regex	jest-each	spdx-license-ids
ansi-regex	jest-environment-jsdom	spdy

ansi-regex	jest-environment-node	spdy-transport
ansi-styles	jest-get-type	specificity
ansi-styles	jest-get-type	Spin.js
ansi-styles	jest-haste-map	sprintf.js
ansi-styles	jest-junit	stable-stringify
anymatch	jest-leak-detector	stackframe
aria-query	jest-matcher-utils	stack-utils
aria-query	jest-message-util	stop-iteration-iterator
array.prototype.findlastindex	jest-pnp-resolver	streamich/memfs
array.prototype.flat	jest-regex-util	strict-uri-encode
array.prototype.flatmap	jest-resolve	string.prototype.matchall
arraybuffer.prototype.slice	jest-resolve-dependencies	string.prototype.trim
array-buffer-byte-length	jest-runner	string.prototype.trimend
array-find	jest-runtime	string.prototype.trimstart
array-flatten	jest-snapshot	string-argv
array-includes	jest-transform-stub	string-length
array-union	jest-util	string-replace-loader
array-union	jest-validate	string-width
array-union	jest-worker	string-width
array-uniq	jQuery	Strip ANSI
arrify	jQuery UI	Strip ANSI
astral-regex	jquery.flot	Strip ANSI
ast-types-flow	jquery-jeditable	strip-bom
async-function	jQuery-Timepicker-Addon	strip-bom
asynckit	js-cookie	strip-comments
at-least-node	jsdom	strip-final-newline
atob	jsesc	strip-indent
Autoprefixer	jsesc	strip-json-comments
available-typed-arrays	jshttp statuses	stylehacks
axe-core	jshttp statuses	stylelint
axios	jshttp statuses	stylelint-config-prettier
axobject-query	jshttp/content-type	stylelint-config-recommended
babel-loader	jshttp/fresh	stylelint-config-standard
babel-plugin-istanbul	JSHTTP's negotiator	stylelint-prettier
babel-plugin-macros	JSHTTP's negotiator	style-search
babel-plugin-polyfill-corejs2	jsjest	stylis
babel-plugin-polyfill-corejs3	json5	supports-color
babel-plugin-polyfill-corejs3	json5	supports-color
babel-plugin-polyfill-regenerator	json-parse-even-better-errors	supports-color
babel-preset-current-node-syntax	json-schema-traverse	supports-color
babel-runtime	json-schema-traverse	supports-hyperlinks

babel-traverse	js-tokens	supports-preserve-symlinks-flag
backbone js framework	js-tokens	svgo
Backbone.Validation	jsx-ast-utils	svgo
balanced-match	js-yaml	svg-tags
balanced-match	js-yaml	symbol-observable
base16	keyanzhang/identity-obj-proxy	symbol-tree
big.js	kind-of	tablesorter jQuery plugin
binary-extensions	klaw	tapable
body-parser	klaw-sync	tapable
bonjour-service	kleur	tapjs/signal-exit
boolbase	klona	terminal-link
brace-expansion	known-css-properties	terser
browserify/resolve	language-subtag-registry	terser-webpack-plugin
browserify/resolve	language-tags	test-exclude
browser-process-hrtime	launch-editor	text-table
browserslist	LessCss	throat
btoa	less-loader	through
buffer-from	leven	thunky
bytes.js	levn	timsort
call-bind	lilconfig	tiny-invariant
call-bind-apply-helpers	lines-and-columns	tiny-warning
call-bound	lint-staged	tmpl
callsites	listr2	toastr
camelcase	ljharb/object.assign	toidentifier
camelcase	loader-runner	tomas/needle
camelcase	locate-path	to-regex-range
camelcase-keys	locate-path	tough-cookie
caniuse-api	Lodash	tr46
caniuse-lite	lodash._getnative	tree-kill
carhartl/jquery-cookie	lodash.curry	trim-newlines
Chalk	lodash.debounce	tsconfig-paths
Chalk	lodash.isarguments	Two-Screen/stable
Chalk	lodash.isarray	type-check
char-regex	lodash.isequal	typed-array-buffer
chokidar	lodash.isplainobject	typed-array-byte-length
chrisdickinson/raf	lodash.keys	typed-array-byte-offset
ci-info	lodash.memoize	typed-array-length
ci-info	lodash.merge	typedarray-to-buffer
cjs-module-lexer	lodash.truncate	type-detect
classes	lodash.uniq	type-fest
classnames	lodash-es	type-fest

classnames	log-update	type-fest
clean-css	loose-envify	type-fest
clean-stack	loose-envify	type-fest
cli-cursor	lower-case	type-is
cli-truncate	lz-string	typicode/husky
cli-truncate	make-dir	unbox-primitive
cliui	make-dir	Underscore.js
clone-deep	makeerror	underscore.string
clone-regexp	map-obj	unicode-canonical-property-names-ecmascript
clsx	map-obj	unicode-match-property-ecmascript
clsx	markedjs	unicode-match-property-value-ecmascript
co	mathiasbynens/he	unicode-property-aliases-ecmascript
collect-v8-coverage	mathiasbynens/regenerate	uniqs
colord	math-intrinsics	universalify
colorette	mathml-tag-names	universalify
color-name	mdn-data	unpipe
color-name	mdn-data	update-browserslist-db
color-string	mdn-data	uri-js
combined-stream	media-typer	url-parse
Commander.js	memory-fs	utila
Commander.js	meow	util-deprecate
Commander.js	merge2	utils-merge
Commander.js	merge-descriptors	uuid
compression	Merge-Stream	v8-compile-cache
concurrently	methods	v8-to-istanbul
connect-history-api-fallback	micromatch	validate-npm-package-license
content-disposition	micromatch/braces	value-equal
convert-source-map	Microsofttslib	vary
convert-source-map	Microsofttslib	vendors
copy-anything	Microsofttslib	visionmedia/batch
copy-webpack-plugin	mime	vitch's jScrollPane
core-js	mime-db	void-elements
core-js	mimic-fn	w3c-hr-time
core-js	min-document	w3c-xmlserializer
core-js-pure	mini-css-extract-plugin	warning
core-util-is	minimalistic-assert	watchman
cosmiconfig	minimatch	watchman
cosmiconfig	minimist	watchpack
create-react-app	minimist-options	wbuf
css	min-indent	webidl-conversions

css.escape	Modernizr	webidl-conversions
css-animation	moment/moment	Webpack
css-blank-pseudo	moxystudio/node-cross-spawn	webpack/loader-utils
css-color-names	moxystudio/node-cross-spawn	webpack/loader-utils
css-color-names	ms.js	webpack/webpack-dev-middleware
css-declaration-sorter	ms.js	webpack-cli
cssesc	ms.js	webpack-deadcode-plugin
css-functions-list	multicast-dns	webpack-dev-server
css-has-pseudo	nanoid	webpack-merge
css-loader	natural-compare	webpack-sources
css-minimizer-webpack-plugin	neo-async	webreflection/html-escaper
cssnano	netroy/image-size	websocket-driver-node
cssnano	nice-try	websocket-extensions-node
cssnano-preset-default	no-case	what-input
csso	Node Cookie Parser	whatwg-encoding
csso	nodeca-argparse	whatwg-mimetype
cssom	nodeca-argparse	whatwg-url
cssom	node-commondir	which
css-prefers-color-scheme	node-concat-map	which
CSSselect	node-cookie-signature	which-boxed-primitive
CSSselect	node-dateformat	which-builtin-type
cssstyle	node-deep-equal	which-collection
csstools/cssdb	node-entities	which-typed-array
css-tree	node-entities	whitequark/ipaddr.js
css-tree	node-error-ex	whitequark/ipaddr.js
css-tree	node-forge	word-wrap
csstype	node-fs-extra	wrap-ansi
css-what	node-fs-extra	wrap-ansi
css-what	node-glob	wrappy
D3.js	node-graceful-fs	xmlchars
d3-array	node-html-entities	xml-name-validator
d3-array	node-http-proxy-agent	y18n
d3-collection	node-ignore	yallist
d3-color	node-int64	yallist
d3-contour	nodejs Deprecate	yaml
d3-format	nodejs Deprecate	yannickcr/eslint-plugin-react
d3-geo	nodejs/string_decoder	yargs
d3-hexbin	nodejs-exit	yargs-parser
d3-hierarchy	node-jsonfile	yocto-queue
d3-interpolate	node-jsonfile	Zurb Foundation

11 Appendix C: EDR Sensor Third Party Libraries Obtained From SBOM

None of the listed libraries in this appendix are dynamic libraries. This is a list of third-party libraries that are statically linked into the code as identified within the SBOM.

Boost	librdkafka	RocksDB
Bzip2	lz4	SQLite
gflags	OpenSSL	yara-doc
google-glog	OSQuery	zlib
libarchive	Protobuf	zstd