Check Point “Harmony” Endpoint Security Host Agent - Security Target

Date: 2025-05-06

Created by

Change History

Version	Date	Author	Comment
1.9	2025-05-06	Check Point Software Technologies	N/A

Table of contents

1 ST Introduction

1.1 ST Reference

Title: Check Point "Harmony" Endpoint Security Host Agent - Security Target

Version: v1.9

Author: Check Point Software Technologies Ltd

Evaluation Lab: jtsec Beyond IT Security

Date of publication: 2025-05-06

1.2 TOE Reference

TOE Name: Endpoint Security Host Agent

TOE Developer: Check Point Software Technologies Ltd

TOE Version: vE88.50 (88.50.0220)

1.3 TOE Overview

1.3.1 Introduction

The TOE is an antivirus software which is part of a client-server architecture intended to be used on computers deployed in a corporate network. These computers will be connected to each other and will carry out information processing to a central server that manages this information in a server-centric architecture. This central server, called Harmony Endpoint EPMaaS is in the cloud and is accessed from the Check Point Infinity Portal.

The TOE is intended to be installed in each computer of the network's organization. Once installed, the TOE carries out the communication to the Harmony Endpoint EPMaaS and monitors the security status of the computer where it has been installed and also synchronises this status with the server. Simultaneously, during this synchronisation process, the TOE applies the Harmony Endpoint EPMaaS's security policies.

1.3.2 TOE Type

The TOE is antivirus software deployed as a software agent that resides on a host platform, an advanced protection solution for endpoints and Internet browsers, ensuring complete protection against different threat vectors with prevention, detection and response capabilities. It incorporates integrated management available in the cloud.

1.3.3 TOE Usage & Major Security Features

The TOE has the following major security features:

- **Cryptographic Support and Data Protection**. The TOE has a group of mechanisms intended to protect the user's data and encrypt the communications performed from the computer where the TOE is installed.
- **Security Management**. The TOE uses mechanisms to allow a secure management of its functionalities. This protection is related to the restriction of using any default credentials and avoiding changes to its binary code once it has been installed, nor during its execution.
- **Identification and Authentication**. This feature is intended to provide an authentication mechanism to the communication between the external servers and the TOE itself.
- **Protection of the TSF.** The TOE includes several mechanisms to protect its critical components and functionalities.
- **Trusted Path/Channels.** The product uses certificates in order to establish a secure communication channel by using the TLS protocol and the HTTPS protocol so that all the information managed and transmitted by the product is encrypted establishing a trusted communication channel.

1.3.4 Non-TOE Hardware/Software/Firmware

As mentioned above, the TOE is intended for being used in a client-server architecture. The TOE is installed on client computers to carry out an analysis of the information handled, as well as the processes it executes in the memory of the client computer operating system. It is necessary to consider that these client computers are connected to a cloud server computer that manages all the events on the client computers, providing a customizable and intuitive control panel.

Due to the possibilities existing in a current corporate network, a client computer where the TOE is installed may be physically located in the internal network or it can be installed in a different geographical location, it only needs Internet access to reach the cloud server in order to be managed.

The TOE needs two main elements to operate:

- **Harmony Endpoint EPMaaS.** The purpose of this element is to carry out the Endpoint client management.
- **General-purpose computer with Windows OS.** The TOE is intended to be installed in a Windows Operating System with at least the Windows 10 version in 64-bit. The Windows versions in scope are Windows 10 (20H2, 21H1, 21H2 and 22H2) and Windows 11 (22H2). Those versions are Common Criteria certified.

1.3.5 Non-evaluated security features

As mentioned above, the product is an antivirus software. This type of product is designed for the prevention, detection and suppression of computer viruses.

The product offers the following protection measures:

- **Compliance**. This feature allows enforcing endpoint compliance on multiple checks before users log into the network. It is possible to check the following:
  - Appropriate endpoint security components are installed.
  - Correct OS service packs are installed on the endpoint.
  - Only approved applications are able to run on the endpoint.
  - Appropriate anti-malware product and version is running on the endpoint.
- **Anti-Malware.** Protects clients from known and unknown viruses, worms, Trojan horses, adware, and keystroke loggers.
- **Anti-bot and URL Filtering.** Detects bot-infected machines and blocks bot Command and Control (C&C) communication to prevent bot damage. Provides detailed information about the device affected by the bot activity, about the bot process itself, and other relevant information.
- **Anti-Ransomware, Behavioral Guard and Forensics**. Prevents ransomware attacks. Monitors files and the registry for suspicious processes and network activity and also analyses incidents reported by other components.
- **Threat Emulation and Anti Exploit**. The product includes a sandbox where it is possible to run securely the files managed by a computer to analyze its behaviour before being executed by the computer itself.

The product employs a series of features called capabilities that can be selected during the product's installation. Each capability deploys a group of protection measures. The capabilities that are installed according to the evaluation's scope during the installation of the product are the following:

- Anti-Malware
- Anti-Bot and URL Filtering
- Anti-Ransomware, Behavioural Guard and Forensics
- Threat Emulation and Anti-Exploit
- Compliance

The following capabilities are excluded during the installation of the product:

- Full Disk Encryption
- Media Encryption and Port Protection
- Remote Access VPN
- Firewall and Application Control

According to the specifications of the protection profile concerning the present security target, the features described in this section have not been considered within the scope of the security target. Therefore, these features will not be considered part of the evaluation as they are not considered part of the TOE.

1.4 TOE Description

1.4.1 TOE Logical Scope

1.4.1.1 Cryptographic Support and Data Protection

The TOE supports the use of encryption suites composed of robust cryptographic algorithms to prevent clear access to critical system security parameters and communications established inside and outside the network where the TOE is installed. The TOE performs the TLS certificate validation. If this certificate's validity cannot be performed successfully during the session establishment, the TOE will not accept the certificate or establish the session. The TOE also is able to perform cryptographic operations as encryption or decryption, hashing, signing and Keyed-Hash Message Authentication, related to the use of different cipher suites.

Additionally, the only sensitive data stored by the TOE is the Endpoint uninstall password used to protect the TOE from unauthorized uninstallation. The storage of other credentials and sensitive data is covered by the Operating System where the TOE is installed. The TOE neither implements the use of access credentials and therefore, it is not possible to make changes to its configuration from the computer where the TOE is installed. In fact, the TOE is limited to using the underlying platform's network connectivity for client/server communications and content updates.

The TOE does not transmit Personally Identifiable Information (PII) over the network. Therefore, the user's data privacy and the whole network's privacy where the TOE is installed are preserved.

The TOE uses a cryptographic module called Cryptocore used for implementing the PBKDF2 (using HMAC-SHA-256) function.

The TOE also uses the OpenSSL library to establish communication channels with the servers. The TOE performs the certificate validation and certificate path by using the OpenSSL library during the TLS handshake.

The underlying platform functionality is also used to establish communication channels with some of the servers. The TOE also uses the operating system functionality to validate the certificate and the certificate path in these connections.

The TOE also uses OpenSSL for random number generation, making calls to the operating system functionality to provide the feature.

1.4.1.2 Security Management

The TOE does not use any default credentials when it is installed. The uninstall password and its corresponding salt are stored in the Windows registry. The authentication mechanisms of the underlying platform are used to ensure only authorized users of that platform can gain access to the application and underlying platform functionality. The TOE does not carry out any changes to its binary code once it has been installed nor during its execution. The TOE is an agent installed on a host platform with reduced management's functionality. The TOE only performs the following management functions: product's update, scan for malware, delete infection, restore infection from quarantine, update malware signature database, enable/disable capabilities and update policies deployed by the Harmony Endpoint EPMaaS.

1.4.1.3 Identification and Authentication

The TOE establishes a communication with the external servers using the HTTPS protocol. This implies the use of the TLS security communication protocol, which is responsible for carrying out the authentication between the TOE itself and the servers.

The TOE and the platform use X.509v3 certificates to authenticate the TLS connection with the external servers. Depending if the communication channel is stablished using the TOE implementation or the platform implementation, the validations is as follows respectively:

- The TOE validates the X.509 certificates using the certificate path validation algorithm and uses the Windows Certificate Store to manage the Certificates Authorities.
- The platform validates the X.509 certificates using the certificate path validation algorithm and uses the Windows Certificate Store to manage the Certificates Authorities.

1.4.1.4 Protection of the TSF

The TOE has protection mechanisms to prevent attacks during its execution. Therefore, the capabilities mentioned above have been compiled into the TOE and are provided since the TOE is executed. The TOE also implements a group of protection mechanisms when compiling occurs. The TOE does not request memory mapping to any explicit address. To achieve this, the TOE includes the following compilation flags: ASLR, DEP and GS.

Regarding the software updates, the TOE allows checking the installed version providing this information on the main interface of the TOE (at the bottom right). Moreover, the TOE allows checking the updates manually with the aim of installing new available updates. To perform this action, it is necessary to use the "Update now" option, where it performs this action and also proceeds to apply the available updates if necessary. The TOE also has a mechanism that allows installing software updates when available. This mechanism allows to install the available updates automatically from the web interface of Harmony Endpoint EPMaaS and its corresponding execution on clients where the TOE is installed. In this way, the TOE takes the necessary steps to verify the integrity of the update packages installed. This verification is carried out using SHA-256 hash algorithm.

The TOE only uses documented platform APIS and third-party libraries to prevent the use of components that could present a privacy threat and ensure that technical vulnerabilities are appropriately addressed.

1.4.1.5 Trusted Path/Channels

During the operation of the TOE, transmitted data is encrypted via HTTPS and TLSv1.2. The TOE allows the use of the HTTPS protocol to carry out the communication with the Harmony Endpoint EPMaaS.

The TOE also supports the use of TLSv1.2 protocol for establishing communication with the Harmony Endpoint EPMaaS. By using the TLSv1.2 protocol, the TOE protects the following information:

- Policy downloads.
- TOE updates.
- New virus database download.
- Heartbeat (a periodic client connection to the server).
- Application Control queries.
- Log uploads.

1.4.2 TOE Physical Scope

The TOE is a software-only evaluation running on a Windows OS platform. The following minimum requirements are needed for the underlying platform to ensure the TOE functions as required:

- **Operating System**: Windows 10 or later.
- **RAM:** 2GB.
- **Hard Drive:** 2GB.
- **Format:** *exe* format.
- **File name:** EPS_*<timestamp>*_V88.50.0220.exe
- **Delivery Method:** official website.
- **TOE version:** vE88.50 (88.50.0220)

The following table lists the documents and user's guide necessary to carry out the configuration of the TOE properly:

Document	Version	Format	Delivery Method
Infinity portal Administration Guide	10 July 2024	PDF document	Email on customer request
Harmony Endpoint EPMAAS Administration Guide	09 July 2024	PDF document	Email on customer request
Endpoint Security Clients For Windows User Guide	02 May 2024	PDF document	Email on customer request

2 Conformance Claims

This Security Target and the TOE described are in accordance with the requirements of Common Criteria 3.1R5.

This Security Target claims conformance with the following parts of Common Criteria:

- Conformance with [CC31R5P2] extended.
- Conformance with [CC31R5P3] extended.

The methodology to be used for the evaluation is described in the "Common Evaluation Methodology" of the Common Criteria standard of April 2017, version 3.1 revision 5 with an evaluation assurance level corresponding to the protection profile to which it claims conformance.

This Security Target claims exact conformance with the following protection profile:

- NIAP Protection Profile for Application Software version 1.4, dated 07 October 2021 with exact conformance.

This Security Target claims conformance with the following package:

- NIAP Functional Package for TLS Version 1.1, dated 12 February 2019.

2.1 Conformance Claims Rationale

2.1.1 TOE Type Consistency

The TOE allows interactive or non-interactive communications with other users or applications over a communications channel. These communications include instant messages, email and voice. Therefore, this behaviour is aligned to Use Case 3 of the Protection Profile for Application Software.

2.1.2 Technical Decisions

The following Technical Decisions have been considered for this evaluation:

Technical Decisions	Applicable	Notes
Protection Profile for Application Software v1.4		
TD0624 - Addition of DataStore for Storing and Setting Configuration Options	No	Superseded by TD0747.
TD0626 - FCS_COP.1 Keyed Hash Selections	No	Superseded by TD0717.
TD0628 - Addition of Container Image to Package Format	Yes	Updated FPT_TUD_EXT.2 to allow the possibility of container images.
TD0650 - Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4	No	VPN Client PP-Module has not been considerated.
TD0655 - Mutual authentication in FTP_DIT_EXT.1 for SW App	No	Superseded by TD0743.
TD0659 - Change to Required NIST Curves for FCS_CKM.1/AK	No	Superseded by TD0717.
TD0664 - Testing activity for FPT_TUD_EXT.2.2	Yes	Related to evaluation activity for FPT_TUD_EXT.2.2.
TD0669 - FIA_X509_EXT.1 Test 4 Interpretation	No	Superseded by TD0780.
TD0709 - Number of elements for iterations of FCS_HTTPS_EXT.1	No	Superseded by TD0736.
TD0717 - Format changes for PP_APP_V1.4	Yes	Related to FCS_CKM.1, FCS_CKM.1.1, FCS_COP.1.1 and its iterations.
TD0719 - ECD for PP APP V1.3 and 1.4	Yes	Extended Component Definitions of the PP have been defined.
TD0736 - Number of elements for iterations of FCS_HTTPS_EXT.1	No	Related to FCS_HTTPS_EXT.1/Server
TD0743 - FTP_DIT_EXT.1.1 Selection exclusivity	Yes	Clarification for FTP_DIT_EXT.1.1
TD0747 - Configuration Storage Option for Android	No	The TOE does not run on Android operating systems.
TD0756 - Update for platform-provided full disk encryption	Yes	Related to evaluation activity for FDP_DAR_EXT.1.

TD0780 - FIA_X509_EXT.1 Test 4 Clarification	Yes	Related to evaluation activity for FIA_X509_EXT.1 Test 4.
TD0798 - Static Memory Mapping Exceptions	Yes	Related to evaluation activity for FPT_AEX_EXT.1.1.
TD0815 - Addition of Conditional TSS Activity for FPT_AEX_EXT.1.5	Yes	Related to evaluation activity for FPT_AEX_EXT.1.5.
TD0822 - Correction to Windows Manifest File for FDP_DEC_EXT.1	No	The TOE is not a Windows Universal Application.
TD0823 - Update to Microsoft Windows Exploit Protection link in FPT_AEX_EXT.1.3	Yes	Related to evaluation activity for FPT_AEX_EXT.1.3.
TD0844 - Addition of Assurance Package for Flaw Remediation V1.0 Conformance Claim	No	Flaw Remediation Assurance Package has not been considered.
Functional Package for TLS v1.1		
TD0779 - Updated Session Resumption Support in TLS package V1.1	No	The TOE does not support TLS server functionality.
TD0770 - TLSS.2 connection with no client cert	No	The TOE does not support TLS server functionality.
TD0739 - PKG_TLS_V1.1 has 2 different publication dates	No	The TOE does not support TLS server functionality.
TD0726 - Corrections to (D)TLSS SFRs in TLS 1.1 FP	No	The TOE does not support TLS/DTLS server functionality.
TD0588 - Session Resumption Support in TLS package	No	Superseded by TD0780.
TD0513 - CA Certificate loading	Yes	FCS_TLSC_EXT.1.3 has been updated.
TD0499 - Testing with pinned certificates	Yes	Related to evaluation activity for FCS_TLSC_EXT.1.2.
TD0469 - Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	No	The TOE does not support TLS server functionality.
TD0442 - Updated TLS Ciphersuites for TLS Package	Yes	FCS_TLSC_EXT.1.1 has been updated.

3 Security Problem Definition

This section describes the security aspects of the operational environment and its expected use in said environment. It includes the declaration of the TOE operational environment that identifies and describes:

- The alleged known threats that will be countered by the TOE
- The organizational security policies that the TOE has to adhere to
- The TOE usage assumptions in the suggested operational environment.

3.1 Threats to Security

This section identifies the threats to assets that require protection by the TOE. The threats are defined in terms of assets concerned, attackers and the adverse action that materializes the threat.

T.NETWORK_ATTACK: An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with the application software or alter communications between the application software and other endpoints in order to compromise it.

T.NETWORK_EAVESDROP: An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between the application and other endpoints.

T.LOCAL_ATTACK: An attacker can act through unprivileged software on the same computing platform on which the application executes. Attackers may provide maliciously formatted input to the application in the form of files or other local communications.

T.PHYSICAL_ACCESS: An attacker may try to access sensitive data at rest.

3.2 Assumptions

The assumptions when using the TOE are the following:

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

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

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

4 Security Objectives

The security objectives are high level declarations, concise and abstract of the solution to the problem exposed in the former section, which counteracts the threats and fulfils the security policies and the assumptions. These consist of:

- the security objectives for the operational environment.
- the security objectives for the TOE

4.1 Security objectives for the TOE

The security objectives for the TOE must determine (to the desired extent) the responsibility of the TOE in countering the threats and in enforcing the OSPs. Each objective must be traced back to aspects of identified threats to be countered by the TOE and to aspects of OSPs to be met by the TOE.

O.INTEGRITY: Addressed by: FDP_DEC_EXT.1, FMT_CFG_EXT.1, FPT_AEX_EXT.1 and FPT_TUD_EXT.1

O.QUALITY: Addressed by: FCS_CKM_EXT.1, FCS_RBG_EXT.1, FCS_STO_EXT.1, FDP_DAR_EXT.1, FMT_MEC_EXT.1, FPT_API_EXT.1, FPT_LIB_EXT.1, FTP_DIT_EXT.1, FPT_TUD_EXT.2, FCS_CKM.1/AK., FCS_CKM.2 and FIA_X509_EXT.1.

O.MANAGEMENT: Addressed by: FMT_SMF.1, FPT_IDV_EXT.1, FPT_TUD_EXT.1, FPR_ANO_EXT.1 and FCS_COP.1/Sig

O.PROTECTED_STORAGE: Addressed by: FDP_DAR_EXT.1, FCS_STO_EXT.1, FCS_RBG_EXT.1, FCS_CKM_EXT.1/PBKDF, FCS_CKM.1/SK, FCS_COP.1/SKC, FCS_COP.1/Hash, FCS_COP.1/KeyedHash and FCS_RBG_EXT.2.

O.PROTECTED_COMMS: Addressed by: FTP_DIT_EXT.1, FCS_RBG_EXT.1, FCS_RBG_EXT.2, FCS_CKM_EXT.1, FCS_CKM.2, FCS_HTTPS_EXT.1/Client HTTPS Protocol, FDP_NET_EXT.1, FIA_X509_EXT.1, FCS_CKM.1/AK, FCS_COP.1/SKC, FCS_COP.1/Hash, FCS_COP.1/Sig, FCS_COP.1/KeyedHash and FIA_X509_EXT.2,

4.2 Security objectives for the operational environment

The security objectives for the Operational Environment determine the responsibility of the environment in countering the threats, enforcing the OSPs and upholding the assumptions. Each objective must be traced back to aspects of identified threats to be countered by the environment, to aspects of OSPs to be enforced by the environment and to assumptions to be uphold by the environment.

OE.PLATFORM: The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying operating system and any discrete execution environment provided to the TOE.

OE.PROPER_USER: The user of the application software is not wilfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy.

OE.PROPER_ADMIN: The administrator of the application software is not careless, wilfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.

4.3 Security Objectives Rationale

The following table provides a mapping of security objectives tracing each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective, and each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective. This illustrates that the security objectives counter all threats, the security objectives enforce all OSPs and the security objectives for the operational environment uphold all assumptions.

	O.INTEGRITY	O.QUALITY	O.MANAGEMENT	O.PROTECTED_STORAGE	O.PROTECTED_COMMS	OE.PLATFORM	OE.PROPER_USER	OE.PROPER_ADMIN
T.NETWORK_ATTACK	**X**		X		**X**			
T.NETWORK_EAVESDROP		**X**	**X**		**X**			
T.LOCAL_ATTACK		**X**						
T.PHYSICAL_ACCESS				**X**				
A.PLATFORM						**X**		
A.PROPER_USER							**X**	
A.PROPER_ADMIN								**X**

Table 1 *Security Objectives vs Security Problem Definition*

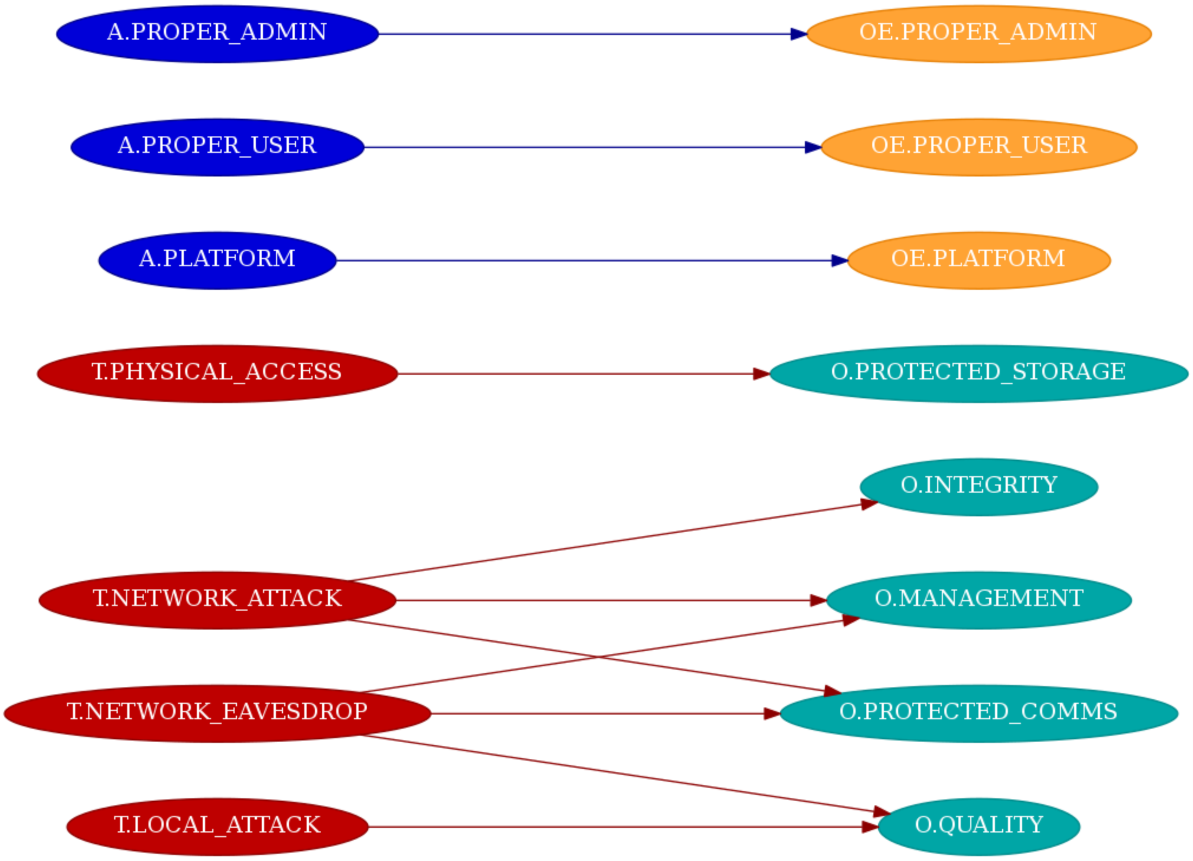


Figure 1 *Mapping of Security Problem Definition to Security Objectives*

4.3.1 Threats

T.NETWORK_ATTACK: The threat T.NETWORK_ATTACK is countered by **O.PROTECTED_COMMS** as this provides for integrity of transmitted data.

The threat T.NETWORK_ATTACK is countered by **O.INTEGRITY** as this provides for integrity of software that is installed onto the system from the network.

The threat T.NETWORK_ATTACK is countered by **O.MANAGEMENT** as this provides for the ability to configure the application to defend against network attack.

T.NETWORK_EAVESDROP: The threat T.NETWORK_EAVESDROP is countered by **O.PROTECTED_COMMS** as this provides for confidentiality of transmitted data.

The objective **O.QUALITY** ensures use of mechanisms that provide protection against network-based attack.

The threat T.NETWORK_EAVESDROP is countered by **O.MANAGEMENT** as this provides for the ability to configure the application to protect the confidentiality of its transmitted data.

T.LOCAL_ATTACK: The objective **O.QUALITY** protects against the use of mechanisms that weaken the **[TOE]** with regard to attack by other software on the platform.

T.PHYSICAL_ACCESS: The objective **O.PROTECTED_STORAGE** protects against unauthorized attempts to access physical storage used by the **[TOE]**.

The following table maps the threats of the security problem established to the security objectives of the TOE and the security objectives of the operational environment.

Threats	Security Objectives
T.NETWORK_ATTACK	O.PROTECTED_COMMS O.INTEGRITY O.MANAGEMENT
T.NETWORK_EAVESDROP	O.PROTECTED_COMMS O.QUALITY O.MANAGEMENT
T.LOCAL_ATTACK	O.QUALITY
T.PHYSICAL_ACCESS	O.PROTECTED_STORAGE

Table 2 Threats vs Security Objectives

4.3.2 Assumptions

A.PLATFORM: The operational environment objective OE.PLATFORM is realized through A.PLATFORM.

A.PROPER_USER: The operational environment objective OE.PROPER_USER is realized through A.PROPER_USER.

A.PROPER_ADMIN: The operational environment objective OE.PROPER_ADMIN is realized through A.PROPER_ADMIN.

The following table maps the assumptions of the problem established to the security objectives of the TOE and the security objectives of the operational environment.

Assumptions	Security Objectives
A.PLATFORM	OE.PLATFORM
A.PROPER_USER	OE.PROPER_USER
A.PROPER_ADMIN	OE.PROPER_ADMIN

Table 3 Assumptions vs Security Objectives for the Operational Environment

5 Extended Components Definition

5.1 Class FCS: Cryptographic support

The TSF may employ cryptographic functionality to help satisfy several high-level security objectives. These include (but are not limited to): identification and authentication, non-repudiation, trusted path, trusted channel and data separation. This class is used when the TOE implements cryptographic functions, the implementation of which could be in hardware, firmware and/or software.

The FCS class is composed of two families: FCS_CKM and FCS_COP. The FCS_CKM family addresses the management aspects of cryptographic keys, while the FCS_COP family is concerned with the operational use of those cryptographic keys.

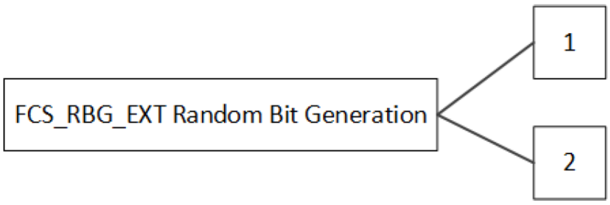
Extension according to Protection Profile for Application Software.

5.1.1 Random Bit Generation (FCS_RBG_EXT)

Family behavior

Family included according to Protection Profile for Application Software.

Component levelling



Management: FCS_RBG_EXT.1, FCS_RBG_EXT.2

There are no management activities foreseen.

Audit: FCS_RBG_EXT.1, FCS_RBG_EXT.2

There are no auditable events foreseen.

FCS_RBG_EXT.1: Random Bit Generation Services

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FCS_RBG_EXT.1.1: ***The application shall [selection: use no DRBG functionality, invoke platform-provided DRBG functionality, implement DRBG functionality] for its cryptographic operations.***

FCS_RBG_EXT.2: Random Bit Generation from Application

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FCS_RBG_EXT.2.1: ***The application shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)]***

FCS_RBG_EXT.2.2: ***The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [selection: a software-based noise source, a hardware-based noise source, no other noise source] with a minimum of [selection: 128 bits, 256 bits] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.***

5.1.2 Cryptographic Key Management (FCS_CKM_EXT)

Family behavior

Family creation according to Protection Profile for Application Software.

Component levelling

FCS_CKM_EXT Cryptographic Key Management
1

Management: FCS_CKM_EXT.1

There are no management activities foreseen.

Audit: FCS_CKM_EXT.1

There are no auditable events foreseen.

FCS_CKM_EXT.1: Cryptographic Key Generation Services

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FCS_CKM_EXT.1.1: ***The application shall [selection: generate no asymmetric cryptographic keys, invoke platform-provided functionality for asymmetric key generation, implement asymmetric key generation]***

5.1.3 Storage of Credentials (FCS_STO_EXT)

Family behavior

Family created according to Protection Profile for Application Software.

Component levelling

Component created according to Protection Profile for Application Software.

Management: FCS_STO_EXT.1

There are no management activities foreseen.

Audit: FCS_STO_EXT.1

There are no auditable events foreseen.

FCS_STO_EXT.1: Storage of Credentials

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FCS_STO_EXT.1.1: ***The application shall [selection: not store any credentials, invoke the functionality provided by the platform to securely store [assignment: list of credentials], implement functionality to securely store [assignment: list of credentials] according to [selection: FCS_COP.1/ SKC, FCS_CKM_EXT.1/PBKDF]] to non-volatile memory.***

5.1.4 HTTPS Protocol (FCS_HTTPS_EXT)

Family behavior

No description.

Component levelling

No description.

Management: FCS_HTTPS_EXT.1/Client

There are no management activities foreseen.

Audit: FCS_HTTPS_EXT.1/Client

There are no auditable events foreseen.

FCS_HTTPS_EXT.1/Client: HTTPS Protocol

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FCS_HTTPS_EXT.1.1/Client: ***The application shall implement the HTTPS protocol that complies with RFC 2818.***

FCS_HTTPS_EXT.1.2/Client: ***The application shall implement HTTPS using TLS as defined in the Functional Package for TLS.***

FCS_HTTPS_EXT.1.3/Client: ***The application shall [selection: not establish the application-initiated connection, notify the user and not establish the user-initiated connection, notify the user and request authorization to establish the user-initiated connection] if the peer certificate is deemed invalid.***

5.1.5 TLS Protocol (FCS_TLS_EXT)

Family behavior

Family created according to Functional Package for Transport Layer Security (TLS).

Component levelling

FCS_TLS_EXT: TLS Protocol
1

Management: FCS_TLS_EXT.1

There are no management activities foreseen.

Audit: FCS_TLS_EXT.1

There are no auditable events foreseen.

FCS_TLS_EXT.1: TLS Protocol

Hierarchical to:

No other components.

Dependencies:

No dependencies.

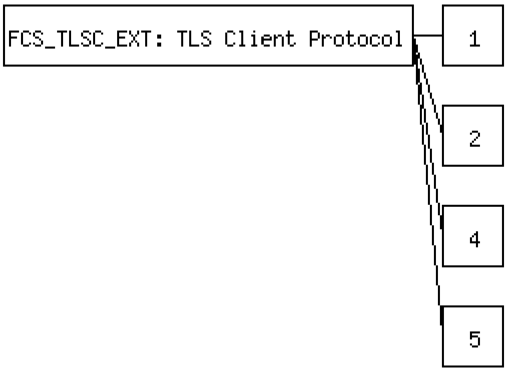
FCS_TLS_EXT.1.1: ***The product shall implement [selection: TLS as a client, TLS as a server, DTLS as a client, DTLS as a server]***

5.1.6 TLS Client Protocol (FCS_TLSC_EXT)

Family behavior

Family created according to Functional Package for Transport Layer Security (TLS).

Component levelling



Management: FCS_TLSC_EXT.1

There are no management activities foreseen.

Audit: FCS_TLSC_EXT.1

There are no auditable events foreseen.

FCS_TLSC_EXT.1: TLS Client Protocol

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FCS_TLSC_EXT.1.1: ***The product shall implement TLS 1.2 (RFC 5246) and [selection: TLS 1.1 (RFC 4346), no earlier TLS versions] as a client that support the cipher suites [selection: TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289] and also supports functionality for [selection: mutual authentication,, session renegotiation,, none]***

FCS_TLSC_EXT.1.2: ***The product shall verify that the presented identifier matches the reference identifier according to RFC 6125.***

FCS_TLSC_EXT.1.3: ***The product shall not establish a trusted channel if the server certificate is invalid [selection: with no exceptions, except when override is authorized]***

5.2 Class FDP: User data protection

This class contains families specifying requirements related to protecting user data. FDP is split into four groups of families (listed below) that address user data within a TOE, during import, export, and storage as well as security attributes directly related to user data.

The families in this class are organised into four groups:

- User data protection security function policies:

- Forms of user data protection:
- Off-line storage, import and export:
- Inter-TSF communication:

Extension according to the Protection Profile for Application Software.

5.2.1 Access to Platform Resources (FDP_DEC_EXT)

Family behavior

Behaviour created according to Protection Profile for Application Software.

Component levelling

FDP_DEC_EXT: Access to Platform Resources
1

Component created according to Protection Profile for Application Software.

Management: FDP_DEC_EXT.1

There are no management activities foreseen.

Audit: FDP_DEC_EXT.1

There are no auditable events foreseen.

FDP_DEC_EXT.1: Access to Platform Resources

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FDP_DEC_EXT.1.1: ***The application shall restrict its access to [selection: no hardware resources, network connectivity, camera, microphone, location services, NFC, USB, Bluetooth, [assignment: list of additional hardware resources]]***

FDP_DEC_EXT.1.2: ***The application shall restrict its access to [selection: no sensitive information repositories, address book, calendar, call lists, system logs, [assignment: list of additional sensitive information repositories]]***

5.2.2 Network Communications (FDP_NET_EXT)

Family behavior

Family created according to Protection Profile for Application Software.

Component levelling

Component created according to Protection Profile for Application Software.

Management: FDP_NET_EXT.1

There are no management activities foreseen.

Audit: FDP_NET_EXT.1

There are no auditable events foreseen.

FDP_NET_EXT.1: Network Communications

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FDP_NET_EXT.1.1: ***The application shall restrict network communication to [selection: no network communication, user-initiated communication for [assignment: list of functions for which the user can initiate network communication], respond to [assignment: list of remotely initiated communication], [assignment: list of application-initiated network communication]]***

5.2.3 Encryption Of Sensitive Application Data (FDP_DAR_EXT)

Family behavior

Family created according to Protection Profile for Application Software.

Component levelling

Component created according to Protection Profile for Application Software.

Management: FDP_DAR_EXT.1

There are no management activities foreseen.

Audit: FDP_DAR_EXT.1

There are no auditable events foreseen.

FDP_DAR_EXT.1: Encryption Of Sensitive Application Data

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FDP_DAR_EXT.1.1: ***The application shall [selection: leverage platform-provided functionality to encrypt sensitive data, implement functionality to encrypt sensitive data as defined in the PP-Module for File Encryption, protect sensitive data in accordance with FCS_STO_EXT.1, not store any sensitive data] in non-volatile memory.***

5.3 Class FMT: Security management

This class is intended to specify the management of several aspects of the TSF: security attributes, TSF data and functions. The different management roles and their interaction, such as separation of capability, can be specified.

This class has several objectives:

- management of TSF data, which include, for example, banners;
- management of security attributes, which include, for example, the Access Control Lists, and Capability Lists;
- management of functions of the TSF, which includes, for example, the selection of functions, and rules or conditions influencing the behaviour of the TSF;
- definition of security roles.

Extension according to the Protection Profile for Application Software.

5.3.1 Supported Configuration Mechanism (FMT_MEC_EXT)

Family behavior

Family creation according to the Protection Profile for Application Software.

Component levelling

Component creation according to Protection Profile for Application Software.

Management: FMT_MEC_EXT.1

There are no management activities foreseen.

Audit: FMT_MEC_EXT.1

There are no auditable events foreseen.

FMT_MEC_EXT.1: Supported Configuration Mechanism

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FMT_MEC_EXT.1.1: ***The application shall [selection: invoke the mechanisms recommended by the platform vendor for storing and setting configuration options, implement functionality to encrypt and store configuration options as defined by FDP_PRT_EXT.1 in the PP-Module for File Encryption].***

5.3.2 Secure by Default Configuration (FMT_CFG_EXT)

Family behavior

Family creation according to Protection Profile for Application Software.

Component levelling

Component creation according to Protection Profile for Application Software.

Management: FMT_CFG_EXT.1

There are no management activities foreseen.

Audit: FMT_CFG_EXT.1

There are no auditable events foreseen.

FMT_CFG_EXT.1: Secure by Default Configuration

Hierarchical to:

No other components.

Dependencies:

No dependencies.

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.***

5.4 Class FPR: Privacy

This class contains privacy requirements. These requirements provide a user protection against discovery and misuse of identity by other users.

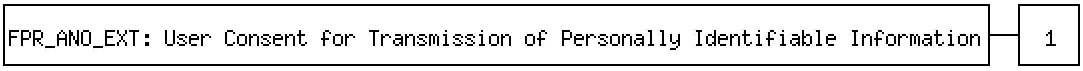
Extension according to Protection Profile for Application Software.

5.4.1 User Consent for Transmission of Personally Identifiable Information (FPR_ANO_EXT)

Family behavior

Family creation according to Protection Profile for Application Software.

Component levelling


FPR_ANO_EXT: User Consent for Transmission of Personally Identifiable Information
1

Component creation according to Protection Profile for Application Software.

Management: FPR_ANO_EXT.1

There are no management activities foreseen.

Audit: FPR_ANO_EXT.1

There are no auditable events foreseen.

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

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FPR_ANO_EXT.1.1: ***The application shall [selection: not transmit PII over a network, require user approval before executing [assignment: list of functions that transmit PII over a network]]***

5.5 Class FPT: Protection of the TSF

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF data. In some sense, families in this class may appear to duplicate components in the FDP class; they may even be implemented using the same mechanisms. However, FDP focuses on user data protection, while FPT focuses on TSF data protection. In fact, components from the FPT class are necessary to provide requirements that the SFPs in the TOE cannot be tampered with or bypassed.

From the point of view of this class, regarding to the TSF there are three significant elements:

- The TSF's implementation, which executes and implements the mechanisms that enforce the SFRs.
- The TSF's data, which are the administrative databases that guide the enforcement of the SFRs.

- The external entities that the TSF may interact with in order to enforce the SFRs.

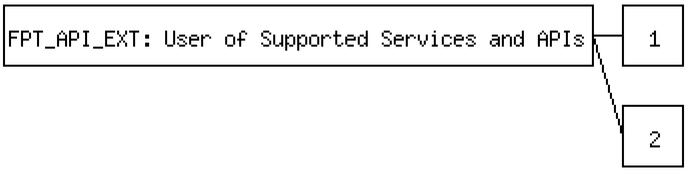
Extension according to Protection Profile for Application Software.

5.5.1 User of Supported Services and APIs (FPT_API_EXT)

Family behavior

Family creation according to Protection Profile for Application Software.

Component levelling



Component creation according to Protection Profile for Application Software.

Management: FPT_API_EXT.1

There are no management activities foreseen.

Audit: FPT_API_EXT.1

There are no auditable events foreseen.

FPT_API_EXT.1: Use of Supported Services and APIs

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FPT_API_EXT.1.1: ***The application shall use only documented platform APIs.***

5.5.2 Anti-Exploitation Capabilities (FPT_AEX_EXT)

Family behavior

Family creation according to Protection Profile for Application Software.

Component levelling

Component creation according to Protection Profile for Application Software.

Management: FPT_AEX_EXT.1

There are no management activities foreseen.

Audit: FPT_AEX_EXT.1

There are no auditable events foreseen.

FPT_AEX_EXT.1: Anti-Exploitation Capabilities

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FPT_AEX_EXT.1.1: ***The application shall not request to map memory at an explicit address except for [assignment: list of explicit exceptions]***

FPT_AEX_EXT.1.2: ***The application shall [selection: not allocate any memory region with both write and execute permissions, allocate memory regions with write and execute permissions for only [assignment: list of functions performing just-in-time compilation]]***

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.***

5.5.3 Integrity for Installation and Update (FPT_TUD_EXT)

Family behavior

Family creation according to Protection Profile for Application Software.

Component levelling

FPT_TUD_EXT: Integrity for Installation and Update
1
2

No description.

Management: FPT_TUD_EXT.1, FPT_TUD_EXT.2

There are no management activities foreseen.

Audit: FPT_TUD_EXT.1, FPT_TUD_EXT.2

There are no auditable events foreseen.

FPT_TUD_EXT.1: Integrity for Installation and Update

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FPT_TUD_EXT.1.1: ***The application shall [selection: provide the ability, leverage the platform] to check for updates and patches to the application software.***

FPT_TUD_EXT.1.2: ***The application shall [selection: 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 [selection: with the platform OS, as an additional software package to the platform OS]***

FPT_TUD_EXT.2: Integrity for Installation and Update

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FPT_TUD_EXT.2.1: ***The application shall be distributed using [selection: the format of the platform-supported package manager, a container image].***

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 configuring settings, output files, and audit/log events.***

FPT_TUD_EXT.2.3: ***The application installation package shall be digitally signed such that its platform can cryptographically verify them prior to installation.***

5.5.4 Use of Third Party Libraries (FPT_LIB_EXT)

Family behavior

Family creation according to Protection Profile for Application Software.

Component levelling

FPT_LIB_EXT: Use of Third Party Libraries
1

Component creation according to Protection Profile for Application Software.

Management: FPT_LIB_EXT.1

There are no management activities foreseen.

Audit: FPT_LIB_EXT.1

There are no auditable events foreseen.

FPT_LIB_EXT.1: Use of Third Party Libraries

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FPT_LIB_EXT.1.1: ***The application shall be packaged with only [assignment: list of third-party libraries]***

5.5.5 Software Identification and Versions (FPT_IDV_EXT)

Family behavior

Family creation according to Protection Profile for Application Software.

Component levelling

Management: FPT_IDV_EXT.1

There are no management activities foreseen.

Audit: FPT_IDV_EXT.1

There are no auditable events foreseen.

FPT_IDV_EXT.1: Software Identification and Versions

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FPT_IDV_EXT.1.1: ***The application shall be versioned with [selection: SWID tags that comply with minimum requirements from ISO/IEC 19770-2:2015, [assignment: other version information]]***

5.6 Class FTP: Trusted path/channels

Families in this class provide requirements for a trusted communication path between users and the TSF, and for a trusted communication channel between the TSF and other trusted IT products. Trusted paths and channels have the following general characteristics:

- The communications path is constructed using internal and external communications channels (as appropriate for the component) that isolate an identified subset of TSF data and commands from the remainder of the TSF and user data.
- Use of the communications path may be initiated by the user and/or the TSF (as appropriate for the component).
- The communications path is capable of providing assurance that the user is communicating with the correct TSF, and that the TSF is communicating with the correct user (as appropriate for the component).

In this paradigm, a trusted channel is a communication channel that may be initiated by either side of the channel, and provides non-repudiation characteristics with respect to the identity of the sides of the channel.

A trusted path provides a means for users to perform functions through an assured direct interaction with the TSF. Trusted path is usually desired for user actions such as initial identification and/or authentication, but may also be desired at other times during a user's session. Trusted path exchanges may be initiated by a user or the TSF. User responses via the trusted path are guaranteed to be protected from modification by or disclosure to untrusted applications.

Extension performed according to Protection Profile Application Software.

5.6.1 Protection of Data in Transit (FTP_DIT_EXT)

Family behavior

Family created according to Protection Profile for Application Software.

Component levelling

FTP_DIT_EXT: Protection of Data in Transit
1

Component created according to Protection Profile for Application Software.

Management: FTP_DIT_EXT.1

There are no management activities foreseen.

Audit: FTP_DIT_EXT.1

There are no auditable events foreseen.

FTP_DIT_EXT.1: Protection of Data in Transit

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FTP_DIT_EXT.1.1: ***The application shall [selection: not transmit any [selection: data, sensitive data], encrypt all transmitted [selection: sensitive data, data] with [selection:*** **HTTPS in accordance with FCS_HTTPS_EXT.1/Client for [assignment: function(s)], HTTPS as a server in accordance with FCS_HTTPS_EXT.1/Server for [assignment: function(s)], HTTPS as a server using mutual authentication in accordance with FCS_HTTPS_EXT.2 for [assignment: function(s)], TLS as a server as defined in the Functional Package for TLS and also supports functionality for [selection: mutual authentication, none] for [assignment: function(s)], TLS as a client as defined in the Functional Package for TLS for [assignment: function(s)], DTLS as a server as defined in the Functional Package for TLS and also supports functionality for [selection: mutual authentication, none] for [assignment: function(s)], DTLS as a client as defined in the Functional Package for TLS for [assignment: function(s)],** ***SSH as defined in the Functional Package for Secure Shell for [assignment: function(s)],*** **IPsec as defined in the PP-Module for VPN Client for [assignment: function(s)]** ***], invoke platform-provided functionality to encrypt all transmitted sensitive data with [selection: HTTPS, TLS, DTLS, SSH] for [assignment: function(s)], invoke platform-provided functionality to encrypt all transmitted data with [selection: HTTPS, TLS, DTLS, SSH]]*** **for [assignment: function(s)]** ***between itself and another trusted IT product.***

5.7 Class FIA: Identification and authentication

Families in this class address the requirements for functions to establish and verify a claimed user identity.

Identification and Authentication is required to ensure that users are associated with the proper security attributes (e.g., identity, groups, roles, security or integrity levels).

The unambiguous identification of authorised users and the correct association of security attributes with users and subjects is critical to the enforcement of the intended security policies. The families in this class deal with determining and verifying the identity of users, determining their authority to interact with the TOE, and with the correct association of security attributes for each authorised user. Other classes of requirements (e.g., User Data Protection, Security Audit) are dependent upon correct identification and authentication of users in order to be effective.

Extension performed according to Protection Profile Application Software.

5.7.1 X.509 Certificate Validation (FIA_X509_EXT)

Family behavior

No description.

Component levelling

Management: FIA_X509_EXT.1, FIA_X509_EXT.2

There are no management activities foreseen.

Audit: FIA_X509_EXT.1, FIA_X509_EXT.2

There are no auditable events foreseen.

FIA_X509_EXT.1: X.509 Certificate Validation

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FIA_X509_EXT.1.1: ***The application shall [selection: invoked platform-provided functionality, implement functionality] in accordance with the following rules:***

a) ***RFC 5280 certificate validation and certificate path validation.***

b) ***The certificate path must terminate with a trusted CA certificate.***

c) ***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.***

d) The application shall validate that any CA certificate includes caSigning purpose in the key usage field

e) ***The application shall validate the revocation status of the certificate using [selection: OCSP as specified in RFC 6960, CRL as specified in RFC 5280 Section 6.3, CRL as specified in RFC 8603, an OCSP TLS Status Request Extension (OCSP stapling) as specified in RFC 6066, OCSP TLS Multi-Certificate Status Request Extension (i.e., OCSP Multi-Stapling) as specified in RF 6961]***

f) ***The application shall validate the extendedKeyUsage field according to the following rules:***

a) *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the EKU field.*

b) *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.*

c) *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.*

d) *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.*

e) *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.*

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

FIA_X509_EXT.1.2: *The application shall treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.*

FIA_X509_EXT.2: Certificate Authentication

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FIA_X509_EXT.2.1: *The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection: HTTPS, TLS, DTLS, SSH, IPsec]*

FIA_X509_EXT.2.2: *When the application cannot establish a connection to determine the validity of a certificate, the application shall [selection: allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate]*

5.8 Extended SARs Definition

5.8.1 Class ALC: Life-cycle support

Class extension according to Protection Profile for Application Software.

5.8.1.1 Timely Security Updates (ALC_TSU_EXT)

Family objectives

Family created according to Protection Profile for Application Software.

Component levelling

ALC_TSU_EXT: Timely Security Updates
1

Component creation according to Protection Profile for Application Software.

ALC_TSU_EXT.1: Timely Security Updates

Hierarchical to:

No other components.

Dependencies:

No dependencies.

Developer action elements.

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.***

Content and representation 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.1.2C: ***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.1.3C: ***The description shall include the mechanisms publicly available for reporting security issues pertaining to the TOE.***

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.***

6 Security Requirements

This section defines the Security functional requirements (SFRs) and the Security assurance requirements (SARs) that fulfill the TOE. Assignment, selection, iteration and refinement operations have been made, adhering to the following conventions:

- Assignments. They appear between square brackets. The word "assignment" is maintained and the resolution is presented in ***boldface, italic and blue color.***
- Selections. They appear between square brackets. The word "selection" is maintained and the resolution is presented in ***boldface, italic and blue color.***
- Iterations. It includes "/" and an "identifier" following requirement identifier that allows to distinguish the iterations of the requirement. Example: FCS_COP.1/XXX.
- Refinements: the text where the refinement has been done is shown ***bold, italic, and light red color.*** Where part of the content of a SFR component has been removed, the removed text is shown in ~~***bold, italic, light red color and crossed out.***~~

The refinements already performed in [PPAPP14] are not identified (e.g., highlighted) here, rather the refined requirements have been copied from [PPAPP14] and any residual operations have been completed herein.

6.1 Security Functional Requirements

The following optional SFRs from Appendix A of PP **[PPAPP14]** have been included in this ST: FCS_CKM.1/SK.

The following selectable SFRs from Appendix B of PP **[PPAPP14]** have been included in this ST because the values associated to each one of them were chosen in the corresponding selections of other SFRs: FCS_CKM.2, FCS_COP.1/SKC, FCS_COP.1/Hash, FCS_COP.1/Sig, FCS_COP.1/KeyedHash, FCS_HTTPS_EXT.1/Client, FCS_TLSC_EXT.1, FCS_TLSC_EXT.5, FIA_X509_EXT.1, FIA_X509_EXT.2, FPT_TUD_EXT.2, FCS_CKM.1/AK and FCS_CKM_EXT.1/PBKDF.

The following selectable SFRs from Appendix B of PP **[PPAPP14]** have not been included in this ST because the values associated to each one of them were not chosen in the corresponding selections of other SFRs: FCS_TLSC_EXT.2, FCS_TLSC_EXT.4.

None of the objective SFRs from Appendix C of PP **[PPAPP14]** have been included in this ST.

6.1.1 FCS: Cryptographic support

6.1.1.1 FCS_CKM.2: Cryptographic Key Establishment

FCS_CKM.2.1 The application shall ***[selection: invoke platform-provided functionality, implement functionality]*** to perform cryptographic keys in accordance with a specified cryptographic key establishment method: ***[selection: [RSA-based key establishment schemes]*** that meet the following: ***RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", [Elliptic curve-based key establishment schemes]*** that meets the following: ***[NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"]]***.

Application Note

The cryptographic key establishment is implemented by the TOE by means of the OpenSSL library in accordance with

- RSA-based key establishment scheme meets the RSA public key encryption and signatures defined in PKCS #1 v2.0 [RFC 2437]. Section 7.2 of RFC 2437 is equivalent to the same version of RFC 8017, Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1.

On the other hand, when the communication is performed by invoking the platform-provided functionality, the cryptographic key establishment is in accordance with:

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

6.1.1.2 FCS_CKM.1/AK: Cryptographic Asymmetric Key Generation

FCS_CKM.1.1/AK The application shall ***[selection: invoke platform-provided functionality, implement functionality]*** to generate ***asymmetric*** cryptographic keys in accordance with a specified cryptographic key generation algorithm ***[selection: [RSA schemes]*** using cryptographic key sizes of ***[2048-bit or greater]*** that meet the following ***[FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3]***, ***[ECC schemes]*** using ***["NIST curves"P-384 and [selection: P-256, P-521]]*** that meet the following: ***[FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4]]***

Application Note

The TOE invokes platform-provided functionality to generate asymmetric cryptographic keys using RSA schemes with key sizes of 2048 and 3072 bits and ECC schemes using NIST curves P-256 and P-384.

This SFR has been included in order to satisfy the dependency with **FCS_CKM.2**. The TOE implements this functionality and is able to invoke the platform for asymmetric key generation. However, this functionality is not used or is available to the end-user as this functionality is not needed to establish a TLS connection with a specific server.

6.1.1.3 FCS_CKM.1/SK: Cryptographic Symmetric key generation

FCS_CKM.1.1/SK The application shall generate symmetric cryptographic keys using a Random Bit Generator as specified ***[assignment: FCS_RBG_EXT.1]*** and specified cryptographic key sizes ***[selection: 128 bit, 256 bit].***

Application Note

This SFR has been included in this Security Target in order to fulfil the missing dependency of **FCS_COP.1/SKC** which is included due to the related TLS functionality implemented by the TOE through the OpenSSL library. As such, this SFR is only applicable for TLS connections initiated through the OpenSSL library, as opposed to the underlying platform.

6.1.1.4 FCS_CKM_EXT.1/PBKDF: Password Conditioning

FCS_CKM_EXT.1.1/PBKDF Refinement: A password/passphrase shall perform ***[assignment: Password-based Key Derivation Functions]*** in accordance with a specified cryptographic algorithm as specified in FCS_COP.1/KeyedHash, with ***[assignment: 2048]*** iterations, and output cryptographic key sizes ***[selection: 256]*** that meet the following: ***[assignment: NIST SP 800-132]***.

FCS_CKM_EXT.1.2/PBKDF The TSF shall generate salts using a RBG that meets FCS_RGB_EXT.1 and with entropy corresponding to the security strength selected for PBKDF in FCS_CKM_EXT.1.1/PBKDF.

6.1.1.5 FCS_COP.1/SKC: Cryptographic operation - Encryption/Decryption

FCS_COP.1.1/SKC The application shall perform [encryption/decryption] in accordance with a specified cryptographic algorithm ***[selection: AES-CBC (as defined in NIST SP 800-38A) mode]*** and cryptographic key sizes ***[selection: 128-bit, 256-bit]***.

6.1.1.6 FCS_COP.1/Hash: Cryptographic operation - Hashing

FCS_COP.1.1/Hash The ***application*** shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm ***[selection: SHA-1, SHA-256]*** and ***message digest*** sizes ***[selection: 160, 256] bits*** that meet the following: [FIPS Pub 180-4].

6.1.1.7 FCS_COP.1/Sig: Cryptographic operation - Signing

FCS_COP.1.1/Sig The ***application*** shall perform [cryptographic signature services (generation and verification)] in accordance with a specified cryptographic algorithm ***[selection: RSA schemes using***

cryptographic key sizes of [2048-bit or greater] that meet the following: [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5]].

6.1.1.8 FCS_COP.1/KeyedHash: Cryptographic operation - Keyed-Hash Message Authentication

FCS_COP.1.1/KeyedHash The ***application*** shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm ***[selection HMAC-SHA-256]*** and ***[selection: SHA-1] with*** key sizes ***[assignment: 160 and 256 bits] and message digest sizes [selection: 256] and [selection: 160] bits*** that meet the following: [*FIPS Pub 198-1 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'*].

6.1.1.9 FCS_RBG_EXT.1: Random Bit Generation Services

FCS_RBG_EXT.1.1 The application shall ***[selection: invoke platform-provided DRBG functionality, implement DRBG functionality]*** for its cryptographic operations.

Application Note

The TOE implements DRBG functionality for its cryptographic operations when the communication is established between the TOE and external servers with the OpenSSL implementation. On the other hand, when communication is established using the platform, the generation of random bits is also provided by platform's functionality.

6.1.1.10 FCS_RBG_EXT.2: Random Bit Generation from Application

FCS_RBG_EXT.2.1 The application shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using ***[selection: CTR_DRBG (AES)].***

FCS_RBG_EXT.2.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and ***[selection: a software-based noise source]*** with a minimum of ***[selection: 256 bits]*** of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

6.1.1.11 FCS_CKM_EXT.1: Cryptographic Key Generation Services

FCS_CKM_EXT.1.1 The application shall ***[selection: invoke platform-provided functionality for asymmetric key generation, implement asymmetric key generation].***

Application Note

The reasoning about the selection chosen in this SFR is directly related to the Application Note defined in **FCS_CKM.1/AK**.

6.1.1.12 FCS_STO_EXT.1: Storage of Credentials

FCS_STO_EXT.1.1 The application shall ***[selection: implement functionality to securely store [assignment: endpoint uninstall password]*** according to ***[selection: FCS_CKM_EXT.1/PBKDF]]*** to non-volatile memory.

6.1.1.13 FCS_HTTPS_EXT.1/Client HTTPS Protocol: HTTPS Protocol

FCS_HTTPS_EXT.1.1/Client HTTPS Protocol The application shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2/Client HTTPS Protocol The application shall implement HTTPS using TLS as defined in the Functional Package for TLS.

FCS_HTTPS_EXT.1.3/Client HTTPS Protocol The application shall ***[selection: not establish the application-initiated connection]*** if the peer certificate is deemed invalid.

6.1.1.14 FCS_TLS_EXT.1: TLS Protocol

FCS_TLS_EXT.1.1 The product shall implement ***[selection: TLS as a client].***

6.1.1.15 FCS_TLSC_EXT.1: TLS Client Protocol

FCS_TLSC_EXT.1.1 The product shall implement TLS 1.2 (RFC 5246) and ***[selection: no earlier TLS versions]*** as a client that support the cipher suites ***[selection: TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246,TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 5246,TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246]*** and also supports functionality for ***[selection: none].***

FCS_TLSC_EXT.1.2 The product shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3 The product shall not establish a trusted channel if the server certificate is invalid ***[selection: with no exceptions].***

6.1.2 FMT: Security management

6.1.2.1 FMT_SMF.1: Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: ***[selection: [assignment: product's update, scan for malware, delete infection, restore infection from quarantine, update Anti-Malware database and enable/disable capabilities and update the following policies: Anti-Malware, Compliance, Anti-Bot, Anti-Ransomware, Behavioral Guard and Forensics]]***.

6.1.2.2 FMT_MEC_EXT.1: Supported Configuration Mechanism

FMT_MEC_EXT.1.1 The application shall ***[selection: invoke the mechanisms recommended by the platform vendor for storing and setting configuration options]***.

6.1.2.3 FMT_CFG_EXT.1: Secure by Default Configuration

FMT_CFG_EXT.1.1 The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

Application Note

The TOE does not install with any default credentials. Rather, it is available to any user logged into the platform.

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.

6.1.3 FDP: User data protection

6.1.3.1 FDP_DEC_EXT.1: Access to Platform Resources

FDP_DEC_EXT.1.1 The application shall restrict its access to ***[selection: network connectivity].***

FDP_DEC_EXT.1.2 The application shall restrict its access to ***[selection: system logs].***

6.1.3.2 FDP_NET_EXT.1: Network Communications

FDP_NET_EXT.1.1 The application shall restrict network communication to ***[selection: [assignment: policy updates, scan commands, virus definition updates, update status information and detection events to Harmony Endpoint EPMaaS]].***

6.1.3.3 FDP_DAR_EXT.1: Encryption Of Sensitive Application Data

FDP_DAR_EXT.1.1 The application shall ***[selection: protect sensitive data in accordance with FCS_STO_EXT.1]*** in non-volatile memory.

6.1.4 FPR: Privacy

6.1.4.1 FPR_ANO_EXT.1: User Consent for Transmission of Personally Identifiable Information

FPR_ANO_EXT.1.1 The application shall ***[selection: not transmit PII over a network].***

6.1.5 FPT: Protection of the TSF

6.1.5.1 FPT_API_EXT.1: Use of Supported Services and APIs

FPT_API_EXT.1.1 The application shall use only documented platform APIs.

6.1.5.2 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 ***[assignment: none].***

FPT_AEX_EXT.1.2 The application shall ***[selection: not allocate any memory region with both write and execute permissions].***

FPT_AEX_EXT.1.3 The application shall be compatible with security features provided by the platform vendor.

FPT_AEX_EXT.1.4 The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.

FPT_AEX_EXT.1.5 The application shall be built with stack-based buffer overflow protection enabled.

6.1.5.3 FPT_TUD_EXT.1: Integrity for Installation and Update

FPT_TUD_EXT.1.1 The application shall ***[selection: provide the ability]*** to check for updates and patches to the application software.

FPT_TUD_EXT.1.2 The application shall ***[selection: provide the ability]*** to query the current version of the application software.

FPT_TUD_EXT.1.3 The application shall not download, modify, replace or update its own binary code.

FPT_TUD_EXT.1.4 Application updates shall be digitally signed such that the application platform can cryptographically verify them prior to installation.

FPT_TUD_EXT.1.5 The application is distributed ***[selection: as an additional software package to the platform OS].***

6.1.5.4 FPT_TUD_EXT.2: Integrity for Installation and Update

FPT_TUD_EXT.2.1 The application shall be distributed using the format of the platform-supported package manager.

FPT_TUD_EXT.2.2 The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuring settings, output files, and audit/log events.

FPT_TUD_EXT.2.3 The application installation package shall be digitally signed such that its platform can cryptographically verify them prior to installation.

6.1.5.5 FPT_LIB_EXT.1: Use of Third Party Libraries

FPT_LIB_EXT.1.1 The application shall be packaged with only ***[assignment: with the third-party library list included in the following application note].***

Application Note

The third-party library list is included in the Annex1 document attached in conjunction to the current Security Target.

6.1.5.6 FPT_IDV_EXT.1: Software Identification and Versions

FPT_IDV_EXT.1.1 The application shall be versioned with ***[selection: [assignment: Check Point internal versioning control]].***

6.1.6 FTP: Trusted path/channels

6.1.6.1 FTP_DIT_EXT.1: Protection of Data in Transit

FTP_DIT_EXT.1.1 The application shall ***[selection: encrypt all transmitted [selection: sensitive data] with [selection: HTTPS in accordance with FCS_HTTPS_EXT.1/Client for [assignment: communications with external servers], TLS as a client as defined in the Functional Package for TLS for [assignment: communications with external servers], invoke platform-provided functionality to encrypt all transmitted sensitive data with [selection: HTTPS, TLS] for [assignment: communications with external servers]]*** between itself and another trusted IT product.

Application Note

The reason about the options chosen in this SFR is related to the establishment of two different communication channels by the TOE. One channel consists of communications using the TOE implementation. In this case, the TLS1.2 protocol is used to establish the communication channel. This functionality is implemented by the product itself.

Another channel consists of communications using platform implementation. For the establishment of this channel, the product relies on operating system functionalities, making use of the HTTPS and TLS protocols and the cipher suites allowed by Windows.

6.1.7 FIA: Identification and authentication

6.1.7.1 FIA_X509_EXT.1: X.509 Certificate Validation

FIA_X509_EXT.1.1 The application shall ***[selection: invoked platform-provided functionality]*** in accordance with the following rules:

a) RFC 5280 certificate validation and certificate path validation.

b) The certificate path must terminate with a trusted CA certificate.

c) 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.

d) The application shall validate that any CA certificate includes caSigning purpose in the key usage field

e) The application shall validate the revocation status of the certificate using ***[selection: CRL as specified in RFC 5280 Section 6.3].***

f) The application shall validate the extendedKeyUsage field according to the following rules:

a) 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.

b) Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.

c) Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.

d) S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the extendedKeyUsage field.

e) OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

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

FIA_X509_EXT.1.2 The application shall treat a certificate as a CA certificate only if the basicConstraints extension is present and the CA flag is set to TRUE.

Application Note

The options selected in this SFR have been chosen according to the following criteria:

- The communication channel comprised between the TOE itself and the external servers is validated by invoking the platform's provided functionality and the TOE implemented functionality depending of the communication channel established.

6.1.7.2 FIA_X509_EXT.2: Certificate Authentication

FIA_X509_EXT.2.1 The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for ***[selection: HTTPS, TLS].***

FIA_X509_EXT.2.2 When the application cannot establish a connection to determine the validity of a certificate, the application shall ***[selection: not accept the certificate].***

6.2 Security Assurance Requirements

The development and the evaluation of the TOE shall be done in accordance with the protection profile to which it claims conformance. The SARs selected are those of the application software protection profile this ST claims conformance. They provide the expected assurance for the evaluated TOE type.

The following table shows the assurance requirements by reference the individual components in [CC31R5P3]:

Assurance Class	Assurance Components
ASE: Security Target evaluation	ASE_CCL.1: Conformance claims ASE_ECD.1: Extended components definition ASE_INT.1: ST introduction ASE_TSS.1: TOE summary specification ASE_SPD.1: Security problem definition ASE_OBJ.2: Security objectives ASE_REQ.2: Derived security requirements
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE ALC_CMS.1: TOE CM coverage ALC_TSU_EXT.1: Timely Security Updates

Assurance Class	Assurance Components
ADV: Development	ADV_FSP.1: Basic functional specification
AGD: Guidance documents	AGD_OPE.1: Operational user guidance AGD_PRE.1: Preparative procedures
ATE: Tests	ATE_IND.1: Independent testing - conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability survey

Table 4 *Security Assurance Requirements*

6.3 Security Requirements Rationale

6.3.1 Necessity and sufficiency analysis

SFR / TOE Security Objective	O.INTEGRITY	O.QUALITY	O.MANAGEMENT	O.PROTECTED_STORAGE	O.PROTECTED_COMMS
FMT_SMF.1			X		
FCS_CKM.2					X
FCS_COP.1/SKC				X	
FCS_COP.1/Hash				X	
FCS_COP.1/Sig			X		
FCS_COP.1/KeyedHash				X	
FCS_HTTPS_EXT.1/Client HTTPS Protocol					X
FIA_X509_EXT.1					X
FIA_X509_EXT.2					X
FPT_TUD_EXT.2		X			
FCS_RBG_EXT.1				X	X
FCS_CKM_EXT.1					X
FCS_STO_EXT.1				X	
FDP_DEC_EXT.1	X				
FDP_NET_EXT.1					X
FDP_DAR_EXT.1				X	

SFR / TOE Security Objective	O.INTEGRITY	O.QUALITY	O.MANAGEMENT	O.PROTECTED_STORAGE	O.PROTECTED_COMMS
FMT_MEC_EXT.1		X			
FMT_CFG_EXT.1	X				
FPR_ANO_EXT.1			X		
FPT_API_EXT.1		X			
FPT_TUD_EXT.1	X		X		
FPT_LIB_EXT.1		X			
FPT_IDV_EXT.1			X		
FTP_DIT_EXT.1					X
FCS_RBG_EXT.2					X
FCS_CKM.1/(3)FCS_CKM_EXT.1/PBKDF				X	
FCS_TLS_EXT.1					X
FCS_TLSC_EXT.1					X
FPT_AEX_EXT.1	X				
FCS_CKM.1/AK		X			
FCS_CKM.1/SK				X	

Table 5 SFRs / TOE Security Objectives coverage

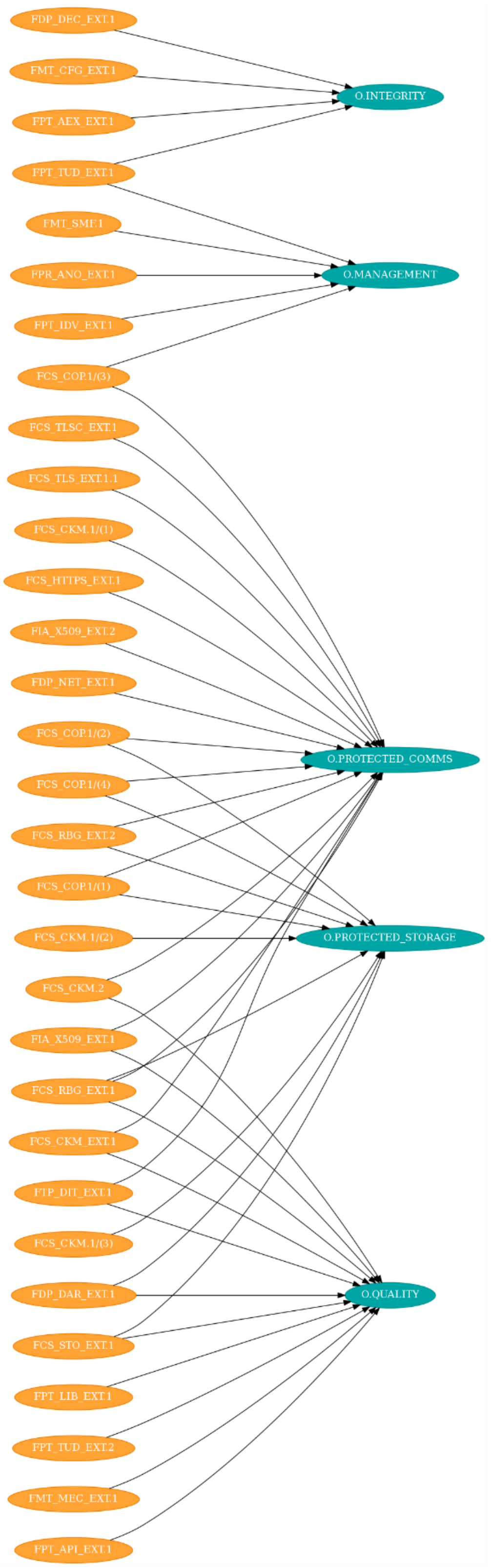


Figure 2 Mapping of SFRs to TOE Security Objectives

6.3.2 Security Requirement Sufficiency

O.INTEGRITY: The PP includes **FDP_DEC_EXT.1** to limit access to platform hardware resources, which limits the methods by which an attacker can attempt to compromise the integrity of the TOE.

The PP includes **FMT_CFG_EXT.1** for the TSF to limit unauthorized access to itself by preventing the use of default authentication credentials and by ensuring that the TOE uses appropriately restrictive platform permissions on its binaries and data.

The PP includes **FPT_AEX_EXT.1** to add complexity to the task of compromising systems by ensuring that application is compatible with security features provided by the platform vendor and that the application implements platform-provided anti-exploitations such as ASLR and stack overflow protection.

The PP includes **FPT_TUD_EXT.1** to ensure that the TOE can be patched and that any updates to the TOE have appropriate integrity protection.

O.QUALITY: The PP supports this objective by allowing **FCS_CKM_EXT.1** to specify that the TSF may rely on platform-provided key generation services.

The PP supports this objective by allowing **FCS_RBG_EXT.1** to specify that the TSF may rely on platform-provided random bit generation services.

The PP supports this objective by allowing **FCS_STO_EXT.1** to specify that the TSF may rely on platform-provided credential storage services.

The PP supports this objective by allowing **FDP_DAR_EXT.1** to specify that the TSF may rely on platform-provided data-at-rest protection services.

The PP includes **FMT_MEC_EXT.1** to ensure that the TOE can use platform services to store and set configuration options.

The PP includes **FPT_API_EXT.1** to require the TOE to leverage platform functionality by using only documented and supported APIs.

The PP includes **FPT_LIB_EXT.1** to ensure that the TOE does not include any unnecessary or unexpected third-party libraries which could present a privacy threat or vulnerability.

The PP supports this objective by allowing **FTP_DIT_EXT.1** to specify that the TSF may rely on platform-provided services to implement trusted communications.

The PP supports this objective by allowing **FCS_CKM.2** to specify that the TSF may rely on platform-provided key establishment services.

The PP supports this objective by allowing **FIA_X509_EXT.1** to specify that the TSF may rely on platform-provided X.509 certificate validation services.

The TSF includes **FPT_TUD_EXT.2** to specify that the TOE may leverage the platform-supported package manager for application distribution and leverages platform-provided mechanisms to remove all traces of itself when removed from the platform system.

The PP supports this objective by allowing FCS_CKM.1/AK to specify that the TSF may rely on platform-provided asymmetric key generation services.

O.MANAGEMENT: The PP includes **FMT_SMF.1** to define the security-relevant management functions that are supported by the TOE.

The PP includes **FPR_ANO_EXT.1** to define how the TSF provides control to the user regarding the disclosure of any PII.

The PP includes **FPT_IDV_EXT.1** to provide a methodology for identifying the TOE versioning.

The PP includes **FPT_TUD_EXT.1** to define how updates to the TOE are deployed and verified.

The PP includes **FCS_COP.1/Sig** to define the mechanism used to verify TOE updates if the TOE implements this functionality rather than the underlying platform.

O.PROTECTED_STORAGE: The PP includes **FCS_RBG_EXT.1** to define whether random bit generation services are implemented by the TSF or the platform. Depending on how data at rest is protected, the TOE may rely on the use of a random bit generator to create keys that are subsequently used for data protection.

The PP includes **FCS_STO_EXT.1** to define the mechanism that the TSF uses or relies upon to protect stored credential data.

The PP includes **FDP_DAR_EXT.1** to define the mechanism that the TSF uses or relies upon to protect sensitive data at rest.

The PP includes **FCS_CKM_EXT.1/PBKDF** to define the password-based key derivation functional that may be used to encrypt stored credential data based on the claims made in **FCS_STO_EXT.1**.

The PP includes **FCS_COP.1/SKC** to define the AES cryptographic algorithm that may be used to encrypt stored credential data based on the claims made in **FCS_STO_EXT.1**.

The PP includes **FCS_COP.1/Hash** to define integrity mechanisms that may be used by the TOE as part of ensuring that data at rest is protected.

The PP includes **FCS_COP.1/KeyedHash** to define HMAC mechanisms that may be used by the TOE as part of ensuring that data at rest is protected.

The PP includes **FCS_RBG_EXT.2** to define the TOE's implementation of random bit generation functionality in the event that the TOE provides this function in support of generating keys that are used for data protection.

The PP includes **FCS_CKM.1/SK** to define the TOE's capability to generate symmetric keys. These keys may subsequently be used to encrypt stored credential data based on the claims made in **FCS_STO_EXT.1**.

O.PROTECTED_COMMS: The PP includes **FCS_RBG_EXT.1** to define whether the random bit generation services used in establishing trusted communications are implemented by the TSF or by the platform.

The PP includes **FCS_CKM_EXT.1** to specify whether the TOE or the platform is responsible for generation of any asymmetric keys that may be used for establishing trusted communications.

The PP includes **FTP_DIT_EXT.1** to define the trusted channels used to protect data in transit, the data that is protected, and whether the trusted channels are implemented by the TSF or the platform.

The PP includes **FCS_CKM.2** to define whether the TSF or the platform performs key establishment for trusted communications.

The PP includes **FCS_COP.1/SKC** to define the symmetric encryption algorithms used in support of trusted communications.

The PP includes **FCS_COP.1/Hash** to define the hash algorithms used in support of trusted communications.

The PP includes **FCS_COP.1/Sig** to define the digital signature algorithms used in support of trusted communications.

The PP includes **FCS_COP.1/KeyedHash** to define the HMAC algorithms used in support of trusted communications.

The PP includes **FCS_RBG_EXT.2** to define the DRBG algorithms used in support of trusted communications.

The PP includes **FCS_HTTPS_EXT.1/Client HTTPS Protocol** to define the TOE's support for the HTTPS trusted communications protocol.

The PP includes **FDP_NET_EXT.1** to define the TOE's usage of network communications, which may include the transmission or receipt of data over a trusted channel.

The PP includes **FIA_X509_EXT.1** to define X.509 certificate validation activities in support of trusted communications.

The PP includes **FIA_X509_EXT.2** to define the trusted communications that X.509 certificate services support, as well as the extent to which trusted communications can be established when using a certificate with unknown validity.

The PP includes **FCS_CKM.1/AK** to define whether the TSF or the platform generates asymmetric keys that are used in support of trusted communications.

6.3.3 SFR Dependency Rationale

The Protection Profile for Application Software contains all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the PP has been approved. Therefore, the *FCS_CKM.*4 missed dependency is fulfilled by this rationale.

Specifically, as no key generation is required for the use of the SHA function, the dependency with FCS_CKM_EXT.1 is not necessary and, therefore, this missing dependency is justified.

6.3.3.1 Table of SFR dependencies

The following table lists the dependencies for each requirement, indicating how they have been satisfied. The abbreviation "h.a." indicates that the dependency has been satisfied by a SFR that is hierarchically above the required dependency.

SFR	Required	Fulfilled	Missing
FMT_SMF.1	None	None	None
FCS_CKM.1/AK	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_CKM.2	FCS_CKM.4
FCS_CKM.1/SK	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_COP.1/SKC	FCS_CKM.4
FCS_CKM_EXT.1/PBKDF	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_COP.1/KeyedHash	FCS_CKM.4
FCS_CKM.2	FCS_CKM.4, [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM_EXT.1]	FCS_CKM.1/AK	FCS_CKM.4
FCS_COP.1/SKC	FCS_CKM.4, [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1/SK]	FCS_CKM.1/SK	FCS_CKM.4
FCS_COP.1/Hash	FCS_CKM.4, [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM_EXT.1]	None	FCS_CKM.4, FCS_CKM_EXT.1
FCS_COP.1/Sig	FCS_CKM.4, [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM_EXT.1]	FCS_CKM.1/AK	FCS_CKM.4
FCS_COP.1/KeyedHash	FCS_CKM.4, [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM_EXT.1]	FCS_CKM_EXT.1/PBKDF	FCS_CKM.4
FCS_HTTPS_EXT.1/Client HTTPS Protocol	None	None	None
FIA_X509_EXT.1	None	None	None
FIA_X509_EXT.2	None	None	None
FPT_TUD_EXT.2	None	None	None
FCS_RBG_EXT.1	None	None	None
FCS_RBG_EXT.2	None	None	None
FCS_CKM_EXT.1	None	None	None
FCS_STO_EXT.1	None	None	None
FDP_DEC_EXT.1	None	None	None

SFR	Required	Fulfilled	Missing
FDP_NET_EXT.1	None	None	None
FDP_DAR_EXT.1	None	None	None
FMT_MEC_EXT.1	None	None	None
FMT_CFG_EXT.1	None	None	None
FPR_ANO_EXT.1	None	None	None
FPT_API_EXT.1	None	None	None
FPT_TUD_EXT.1	None	None	None
FPT_LIB_EXT.1	None	None	None
FPT_IDV_EXT.1	None	None	None
FTP_DIT_EXT.1	None	None	None
FCS_TLS_EXT.1	None	None	None
FCS_TLSC_EXT.1	None	None	None
FPT_AEX_EXT.1	None	None	None

Table 6 SFR Dependencies

6.3.4 SAR Dependency Rationale

6.3.4.1 Table of SAR dependencies

SAR	Required	Fulfilled	Missing
ASE_CCL.1	ASE_INT.1, ASE_ECD.1, ASE_REQ.1	ASE_INT.1, ASE_ECD.1, ASE_REQ.2 (hierarchically above ASE_REQ.1)	None
ASE_ECD.1	None	None	None
ASE_INT.1	None	None	None
ASE_OBJ.2	ASE_SPD.1	ASE_SPD.1	None
ASE_REQ.2	ASE_OBJ.2, ASE_ECD.1	ASE_OBJ.2, ASE_ECD.1	None
ASE_TSS.1	ASE_INT.1, ASE_REQ.1, ADV_FSP.1	ASE_INT.1, ASE_REQ.2 (hierarchically above ASE_REQ.1), ADV_FSP.1	None
ALC_CMC.1	ALC_CMS.1	ALC_CMS.1	None
ALC_CMS.1	None	None	None
ADV_FSP.1	None	None	None
AGD_OPE.1	ADV_FSP.1	ADV_FSP.1	None
AGD_PRE.1	None	None	None

SAR	Required	Fulfilled	Missing
ATE_IND.1	ADV_FSP.1, AGD_OPE.1, AGD_PRE.1	ADV_FSP.1, AGD_OPE.1, AGD_PRE.1	None
AVA_VAN.1	ADV_FSP.1, AGD_OPE.1, AGD_PRE.1	ADV_FSP.1, AGD_OPE.1, AGD_PRE.1	None
ASE_SPD.1	None	None	None
ALC_TSU_EXT.1	None	None	None

Table 7 SAR dependencies

7 TOE Summary Specification

7.1 TOE Security functionality

7.1.1 Cryptographic Support and Data Protection

The product performs cryptographic operations using three different mechanisms. Each mechanism will be used for a particular purpose and will influence the requirements of the security functions:

- One mechanism used by the product is the OpenSSL library, which will be used to perform cryptographic operations to establish communication channels with the servers (symmetric key generation, asymmetric key generation, encryption, decryption, etc.). This mechanism allows the product to implement some security functionalities listed in the different SFR of the present security target. When the "implement functionality" selection was selected, this mechanism is used to provide the required functionality.
- Another mechanism consists of invoking the functionality of the operating system to perform cryptographic operations. The operating system functionality is used to establish secure communication channels with external servers, which the TOE establishes a communication. For the communication purpose of these channels, the TOE will use the operating system's functionality for the cryptographic functionality required.
- Finally, the TOE has a proprietary cryptographic module called CryptoCore. This module is mainly used to protect the product's uninstallation password by using the PBKDF2 function.

The TOE implements DRBG functionality for the communications established using the OpenSSL implementation. On the other hand, when communication is established using the platform's functionality, the platforms performs internally the generation of random bits. This is addressed by **FCS_RBG_EXT.1**. The TOE performs all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using CTR_DRBG (AES) with a minimum of 256 bits of entropy. This is addressed by **FCS_RBG_EXT.2.**

The TOE performs the generation of asymmetric cryptographic keys to establish any communication. If communication is done with OpenSSL, the functionality of the OpenSSL library is used to generate the necessary asymmetric keys. In the case of stablishing the communication channel using platform implementation, the functionality of the platform (Windows Operating System) is invoked to generate the asymmetric keys. This is addressed by **FCS_CKM_EXT.1**.

Both situations described in the generation of asymmetric cryptographic keys are carried out using RSA schemes using cryptographic key sizes of 2048 bits and 3072 bits when the OpenSSL library is used. The TOE invokes platform-provided functionality to generate asymmetric cryptographic keys using RSA schemes with key sizes of 2048 and 3072 bits and ECC schemes using NIST curves P-256, P-384 and P-521. This SFR has been included in order to satisfy the dependency with **FCS_CKM.2**. The TOE implements this functionality and is able to invoke the platform for asymmetric key generation. However, this functionality is not used or is available to the end-user as this functionality is not needed to establish a TLS connection with a specific server. This is addressed by **FCS_CKM.1/AK**.

The TOE generates symmetric cryptographic keys using the internal Random Bit Generator (DRBG) implemented by the OpenSSL library with 128 bits and 256 bits cryptographic key sizes. According to **FCS_RBG_EXT.2.2**, the minimum entropy of the DRBG matches with the greatest security strength, that is, the strength claimed in **FCS_COP.1/SKC** (256 bits) and with the strength claimed in the own cryptographic symmetric key generation. This is addressed by **FCS_CKM.1/SK.**

The TOE implements functionality to securely store the Endpoint Uninstall password. This is addressed by **FCS_STO_EXT.1**. The uninstall password is protected by using PBKDF2 (Password-based Key Derivation Functions) and its corresponding output is stored in the Windows Operating System registry. The PBKDF2 function is performed by the CryptoCore module of the TOE which relies on **FCS_COP.1/KeyedHash** to provide HMAC-SHA-256 with key size of 256. The TOE conditions the uninstall password using a PBKDF2 (using HMAC-SHA-256) function that meets NIST SP 800-132 and RFC 2898 with 2048 iterations to generate a 256-bit KEK to protect the password. All salts are generated invoking platform-provided DRBG functionality (using the *CryptGenRandom* function) as specified in **FCS_RBG_EXT.1**. The password/passphrase is encoded in ASCII, concatenated with the salt and used as input to the CryptoCore module. The output provided by the mentioned API is scrambled and obfuscated through a group of byte-level operations. This is addressed by **FCS_CKM_EXT.1/PBKDF**.

According to the entropy source, OpenSSL makes use of the USE_BCRYPTGENRANDOM function for random number generation. The entropy is obtained by considering the following parameters: running processes, threads, processor, selected window, modules, heap addresses, process identifier, memory usage and others. For more information on the entropy source used by OpenSSL, the analysis of the *rand_win.c* file is encouraged, which will allow to understand in sufficient detail how the entropy source works when running OpenSSL on the Windows operating system. OpenSSL is an open-source software and the mentioned file can be located in GitHub web site in its *OpenSSL_1_1_1-stable branch*.

OpenSSL performs a set of health tests to ensure the proper operation of the entropy source used. For more information on the tests used, it is recommended to access the *randtest.c* file in the OpenSSL library source code. Module users (the calling applications) shall use entropy sources that meet the security strength required for the random number generation mechanism as shown in [SP 800-90] *Table 2. Definitions for Hash-Based DRBG Mechanisms.* This entropy is supplied by means of callback functions. Those functions must return an error if the minimum entropy strength cannot be met.

The TOE restricts its access to the network connectivity hardware and the system logs needed to establish the communications and store relevant audit events. The TOE allows the generation of customized audit logs. This customization is performed by configuration from Harmony Endpoint EPMaaS. The TOE generates audit logs at the beginning and end of the audit function when the TOE starts or ends its operations. The TOE generates audit logs when any of the following events occur: changes involving the product configuration and events related to threat detection on the computer where the TOE is installed. This is addressed by **FDP_DEC_EXT.1**.

The TOE restricts network communication to policy updates and scans commands as well as antivirus definition updates. The TOE also establishes a communication channel to update the status information and detection events. The network communications are restricted to be sent to the Harmony Endpoint EPMaaS and to the following external servers:

Hostname	Used For
*.epmgmt.checkpoint.com	• Client-Server communication • Load Balancing and Failover Routing purposes • Exporting log to SIEM solutions • Patch management feature in clients • During HEP upgrades Source of files • Server Profiles Feature • Forensic reports upload • Anti-Malware E1 engine and signature updates • Updates for endpoint • Application Control
dl3.checkpoint.com	• SBA Signature updates
gwevents.checkpoint.com	• Statistics collection
ftp-proxy.checkpoint.com	• Uploading CPInfo
teadv.checkpoint.com	• US-DHS and EU compliant Anti-Malware engine and signature updates
updates.checkpoint.com	• SBA Signature updates
cws.checkpoint.com	• Anti-Bot URL reputation
sc1.checkpoint.com	• Retrieve URL for bot detection server
secureupdates.checkpoint.com	• Signatures database updates
sophosxl.com	• Live Protection (E87.50 and higher)
s.sophosxl.net	• Live Protection
Region dependent connections: ASIASOUTHEAST • datatubev2prodsoutheasta.blob.core.windows.net • datatubeprodsoutheastasi.blob.core.windows.net	• Threat Hunting data upload

Hostname	Used For
• datatube-prod.azurewebsites.net AUSTRALIAEAST • datatubev2prodaustraliae.blob.core.windows.net • datatubeprodaustraliaeas.blob.core.windows.net • datatube-prod.azurewebsites.net CANADACENTRAL • datatubev2prodcanadacent.blob.core.windows.net • datatubeprodcanadacentra.blob.core.windows.net • datatube-prod.azurewebsites.net EUROPEWEST • dtwesteuropeprod1.blob.core.windows.net • dtwesteuropeprod2.blob.core.windows.net • dtwesteuropeprod3.blob.core.windows.net • datatubeprodwesteurope.blob.core.windows.net • datatube-prod.azurewebsites.net INDIACENTRAL • datatubev2prodcentralind.blob.core.windows.net • datatubeprodcentralindia.blob.core.windows.net • datatube-prod.azurewebsites.net UAENORTH • dtuaenorthprod.blob.core.windows.net • datatube-prod.azurewebsites.net UKWEST • datatubev2produkwest.blob.core.windows.net • datatubeprodukwest.blob.core.windows.net • datatube-prod.azurewebsites.net USEAST • dteastus2prod1.blob.core.windows.net	

Hostname	Used For
• dteastus2prod2.blob.core.windows.net • dteastus2prod3.blob.core.windows.net • datatubeprodeastus2.blob.core.windows.net • datatube-prod.azurewebsites.net	
*.iaas.checkpoint.com	• File reputation service (Horizon IOC) • URL reputation service (Horizon IOC) • Malware service (Horizon IOC) • Interaction with Threat Emulation cloud
rep.checkpoint.com	• ThreatCloud File Reputation service

This is addressed by **FDP_NET_EXT.1**.

The TOE protects the sensitive data in accordance with **FCS_STO_EXT.1**. The sensitive data protected by the TOE is the Endpoint uninstall password by using the PBKDF2 function as described above. This is addressed by **FDP_DAR_EXT.1**.

The TOE does not transmit any PII over the network. This is addressed by **FPR_ANO_EXT.1**.

The TOE establishes two secure communication channels by using different cryptographic key establishment methods:

- TOE TLS implementation, the TOE makes use of TLS and HTTPS to establish the secure communication channel. To achieve this, it makes use of encryption suites that use robust algorithms. The product uses the OpenSSL library (version 1.1.1w) to implement the functionality to perform cryptographic key establishment in accordance with RSA-based key establishment schemes. According to the OpenSSL official documentation, the RSA-based key establishment scheme meets the RSA public key encryption and signatures defined in PKCS #1 v2.0 [RFC 2437]. Section 7.2 of RFC 2437 is equivalent to the same version of RFC 8017, Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1.
- For the other channel, the product relies on operating system functionalities, using the HTTPS and TLS protocols and the cipher suites allowed by the Windows operating system. Therefore, the product uses the operating system libraries to implement the functionality to perform cryptographic key establishment in accordance with RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1" and in accordance with Elliptic curve-based key establishment schemes meeting the NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography".

Both cryptographic key establishment methods are addressed by **FCS_CKM.2**.

The following table defines the endpoints that can be used by each one of the TLS implementations:

TLS implementation	Endpoints
TOE implementation	- *.epmgmt.checkpoint.com - teadv.checkpoint.com - ftp-proxy.checkpoint.com - sc1.checkpoint.com
Platform implementation	- gwevents.checkpoint.com - teadv.checkpoint.com - updates.checkpoint.com - dl3.checkpoint.com - *.iaas.checkpoint.com - rep.checkpoint.com - sophosxl.com - s.sophosxl.net - secureupdates.checkpoint.com - datatubev2prodsoutheasta.blob.core.windows.net - datatubeprodsoutheastasi.blob.core.windows.net - datatube-prod.azurewebsites.net - datatubev2prodaustraliae.blob.core.windows.net - datatubeprodaustraliaeas.blob.core.windows.net - datatube-prod.azurewebsites.net - datatubev2prodcanadacent.blob.core.windows.net - datatubeprodcanadacentra.blob.core.windows.net - datatube-prod.azurewebsites.net - dtwesteuropeprod1.blob.core.windows.net - dtwesteuropeprod2.blob.core.windows.net - dtwesteuropeprod3.blob.core.windows.net - datatubeprodwesteurope.blob.core.windows.net - datatube-prod.azurewebsites.net - datatubev2prodcentralind.blob.core.windows.net - datatubeprodcentralindia.blob.core.windows.net - datatube-prod.azurewebsites.net - dtuaenorthprod.blob.core.windows.net - datatube-prod.azurewebsites.net - datatubev2produkwest.blob.core.windows.net - datatubeprodukwest.blob.core.windows.net - datatube-prod.azurewebsites.net - dteastus2prod1.blob.core.windows.net - dteastus2prod2.blob.core.windows.net - dteastus2prod3.blob.core.windows.net - datatubeprodeastus2.blob.core.windows.net - datatube-prod.azurewebsites.net

Moreover, the product performs the following type of cryptographic operations:

- Encryption/decryption operations capabilities in accordance with the AES-CBC cryptographic algorithm.

- Keyed-hash operation to provide integrity and authentication for each TLS package. In particular, the function used is HMAC with the underlying hashing functions SHA-1 and SHA-256 with message digest sizes 160 and 256 bits.

- Signing operations to perform signature services by using RSA schemes. The RSA scheme is also used for key establishment in TLS communication protocol.

This is addressed by **FCS_COP.1/SKC**, **FCS_COP.1/Hash**, **FCS_COP.1/Sig** and **FCS_COP.1/KeyedHash**.

The TOE implements TLSv1.2 protocol for use in establishing secure connections to external IT entities. The TOE supports the following cipher suites:

- TLS_RSA_WITH_AES_256_CBC_SHA256 included in RFC 5246.

- TLS_RSA_WITH_AES_256_CBC_SHA included in RFC 5246.

- TLS_RSA_WITH_AES_128_CBC_SHA256 included in RFC 5246.

- TLS_RSA_WITH_AES_128_CBC_SHA included in RFC 5246.

The does not establish any type of connection if the peer certificate is deemed invalid. This is addressed by **FCS_HTTPS_EXT.1/Client HTTPS Protocol, FCS_TLS_EXT.1** and **FCS_TLSC_EXT.1**. The product does not support IP address referenced identifiers or support certificate pinning. The wildcards are supported and the reference identifier included in the digital certificate is the Subject Alternative Name (SAN).

The TOE uses the HTTPS protocol in accordance with the RFC2818 standard. Therefore, the product uses the following to establish a connection using HTTPS protocol:

- Initiates a connection to the external server on the appropriate port and then sends the *TLS ClientHello* message to begin the TLS handshake. When the TLS handshake has finished, the product may then initiate the first HTTP request.
- Once a connection closure message is received, the product does not send any additional information and does not reuse the session.
- The product and the external servers use the port 443 to establish a communication channel to use the HTTPS protocol.
- The product is able to validate the server’s certificate to verify that it is validated by a trusted Certificate Authority (CA).

7.1.2 Security Management

The TOE does not provide any Security Relevant configuration options for the software. The software is an agent that installs on the host systems OS. Once installed, the product only allows very limited interaction with the host OS user. This is addressed by **FMT_MEC_EXT.1**.

The TOE does not require any credentials to be configured as the configuration is done from the server. Therefore, once the TOE is installed, it loads the configuration avoiding the need to use credentials. The only credential that the product installs by default is the endpoint uninstall password. This password (the PBKDF2 output which is scrambled as described in section *7.1.1. Cryptographic Support and Data Protection*) and its corresponding salt (without scrambling) are stored in the

Windows registry in Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\CheckPoint\Endpoint Security\UninstPwdHash and Computer\HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\CheckPoint\Endpoint Security\UninstPwdSalt. The value of this credential is defined in the policies from Harmony Endpoint EPMaaS and do not refer to the default credentials of **FMT_CFG_EXT.1** which are related to the protection of the management functionality.

The TOE performs the following management functions: product's update, scan for malware, delete infection, restore infection from quarantine, update malware signature database, enable/disable capabilities , update the Anti-Malware Database and update the following policies: Anti-Malware, Compliance, Anti-Bot, Anti-Ransomware, Behavioral Guard and Forensics. This is addressed by **FMT_SMF.1**.

7.1.3 Identification and Authentication

The TOE establishes two secure communication channels by using different mechanisms to each channel:

- One channel uses the TOE secure channel implementation
- Another channel consists of communication using the platform implemented functionality.

In both cases, the TOE establishes a communication with the external servers using the HTTPS protocol. This implies the use of the TLS security communication protocol, which is responsible for carrying out the authentication between the TOE itself and the server against which communication is established. This authentication is performed by validating the certificate provided by the server using the Windows certificate store, this validation is performed invoking the platform's provided functionality or the TOE implementation depending of the communication channel.

These certificates follow the X.509 standard and to validate this type of certificate, the rules addressed by **FIA_X509_EXT.1** are followed. The Certificate Revocation List is used to validate the revocation status of the certificate.

The TOE performs the certificate validation and certificate path using the OpenSSL library during the TLS handshake when it received the TLS server certificate from an external server through a channel established by the TOE implementation. Similarly, the TOE uses the operating system functionality to validate the certificate and the certificate path in the case of establishing the channel using the platform implementation. The certificate cannot be revoked for the validation process to be successful. If for any reason the TOE or the operating system are unable to determine the validity of a certificate, the certificate will not be accepted. This is addressed by **FIA_X509_EXT.2**.

The X.509 certificates are validated using the certificate path validation algorithm defined in RFC 5280. The main points to consider can be summarized as follows:

- The algorithm requires the CA's public key, the CA's name, and any constraints upon the set of paths that may be validated using this key. The trust anchor is an input to the algorithm.
- The algorithm validates the certificate concerning the current date and time.

- The primary goal of path validation is to verify the binding between a subject-distinguished name or a subject alternative name and subject public key, as represented in the target certificate, based on the public key of the trust anchor.
- Verify the binding between the name and subject public key requires obtaining a sequence of certificates that support that binding. The certification path should satisfy the following conditions (for a sequence of n certificates):
    - For all x in {1, ..., n-1}, the subject of certificate x is the issuer of certificate x+1
    - Certificate 1 is issued by the trust anchor;
    - Certificate n is the certificate to be validated (i.e., the target certificate); and
    - For all x in {1, ..., n}, the certificate was valid at the time in question.
- The algorithm consists of four basic steps: 1. initialization, 2. basic certificate processing, 3. preparation for the next certificate, and 4. wrap-up.

7.1.4 Protection of the TOE

The TOE only uses documented platform APIs according to **FPT_API_EXT.1**. The TOE supports the following APIs: ReleaseBindInfo, SetSecurityDescriptorOwner, CryptGenRandom, SetFileSecurityW, EqualSid, SetServiceStatus, DuplicateToken, RegisterEventSourceW, IsValidSid, InitializeSid, GetSecurityDescriptorLength, LookupAccountNameW, EnumerateTraceGuids, CryptDestroyHash, EnumServicesStatusA, AdjustTokenPrivileges, RegDisablePredefinedCache, OpenEventLogW, ChangeServiceConfigW, CryptGetHashParam, ConvertStringSecurityDescriptorToSecurityDescriptorW, CryptSetKeyParam, RegCreateKeyExA, EnumDependentServicesA, ControlService, GetNumberOfEventLogRecords, LsaFreeMemory, GetSecurityDescriptorOwner, MakeAbsoluteSD, CreateProcessAsUserW, LsaQueryInformationPolicy, SetFileSecurityA, RegDeleteValueA, RegEnumKeyA, CryptReleaseContext, RegUnLoadKeyW, SetSecurityDescriptorDacl, OpenSCManagerA, RegQueryValueExA, ControlTraceW, DeleteService, ChangeServiceConfigA, RegNotifyChangeKeyValue, QueryTraceW, CryptGetUserKey, EnumDependentServicesW, RegConnectRegistryW, AccessCheck, EventWrite, CopySid, AllocateAndInitializeSid, LookupPrivilegeValueW, CryptDestroyKey, CreateWellKnownSid, NotifyChangeEventLog, CryptExportKey, SetSecurityDescriptorGroup, FreeSid, CryptSignHashW, RegEnumValueA, ReportEventA, LockServiceDatabase, LookupPrivilegeNameW, RegFlushKey, InitializeSecurityDescriptor, GetSecurityInfo, RegSetValueExW, OpenTraceW, StartTraceW, CryptCreateHash, RegDeleteValueW, CryptAcquireContextA, RegQueryValueA, QueryServiceConfigW, GetOldestEventLogRecord, RegQueryValueExW, LookupAccountNameA, RegOpenKeyExA, GetSidLengthRequired, ReportEventW, CryptVerifySignatureW, SetSecurityDescriptorSacl, RegSetValueExA, ImpersonateLoggedOnUser, ImpersonateSelf, CryptEncrypt, LookupPrivilegeValueA, CryptHashData, RegisterServiceCtrlHandlerW, ConvertStringSidToSidW, DuplicateTokenEx, ReadEventLogW, OpenServiceA, RegDeleteKeyW, CheckTokenMembership, SetNamedSecurityInfoW, SetNamedSecurityInfoA, RegOpenKeyExW, MapGenericMask, CryptImportKey, OpenProcessToken, LookupAccountSidA, SetEntriesInAclW, RegSetKeySecurity, QueryAllTracesW, InitiateSystemShutdownExW, RegCopyTreeW, GetNamedSecurityInfoA, RegQueryInfoKeyA, RegEnumValueW, LogonUserW, RegLoadKeyW, CreateServiceA, RegisterServiceCtrlHandlerA, IsValidSecurityDescriptor, EnumerateTraceGuidsEx, UnlockServiceDatabase, StartServiceCtrlDispatcherA, ConvertSidToStringSidW, RegDeleteKeyA, StartServiceCtrlDispatcherW, GetSidSubAuthorityCount, NotifyServiceStatusChangeW, RegGetKeySecurity, QueryServiceConfig2W, CryptAcquireContextW, LookupAccountSidW, EnableTraceEx2, InitiateSystemShutdownW, GetNamedSecurityInfoW, RegDeleteKeyExW, RegEnumKeyExW, CryptGetProvParam, SetEntriesInAclA, CryptGenKey, GetAce, InitializeAcl, CloseServiceHandle, AreAllAccessesGranted, GetSecurityDescriptorControl, RegSetKeyValueW, GetSidSubAuthority, RegEnumKeyExA, CryptSetProvParam, ChangeServiceConfig2W, EventRegister, GetEffectiveRightsFromAclA, RegGetValueA, GetSecurityDescriptorSacl, CryptEnumProvidersW, RegCloseKey, QueryServiceStatus, ConvertSecurityDescriptorToStringSecurityDescriptorW, CreateServiceW, GetTokenInformation, RegCreateKeyA, ConvertSidToStringSidA, GetAclInformation, GetFileSecurityW, GetUserNameW, QueryServiceStatusEx, TraceSetInformation, CreateProcessAsUserA, GetSecurityDescriptorDacl, GetUserNameA, OpenServiceW, CryptDecrypt, CloseEventLog, StopTraceW, SetThreadToken, RegEnumKeyW, RegCreateKeyExW, RegDeleteTreeW, AddAccessAllowedAce, SetSecurityInfo, OpenThreadToken, AddAce, RegOpenKeyA, RegGetValueW, GetSidIdentifierAuthority, RegisterServiceCtrlHandlerExW, GetLengthSid, OpenSCManagerW, RegisterEventSourceA, RevertToSelf, MakeSelfRelativeSD, CryptSetHashParam, TreeSetNamedSecurityInfoW, GetSecurityDescriptorGroup, StartServiceW, ProcessTrace, CloseTrace, LsaNtStatusToWinError, LsaOpenPolicy, RegQueryInfoKeyW, DeregisterEventSource, RegQueryValueW, LsaClose, StartServiceA, EventUnregister, CryptDeriveKey, InitCommonControlsEx, ImageList_AddMasked, ImageList_Draw, CertComparePublicKeyInfo, CertEnumSystemStore, CertAddCertificateContextToStore, CryptMsgGetParam, CertFreeCertificateChain, CryptMsgOpenToDecode, CryptMsgClose, CryptDecodeObjectEx, CryptHashCertificate, CryptAcquireCertificatePrivateKey, CertCloseStore, CertNameToStrW, CryptEncodeObjectEx, CryptMsgVerifyCountersignatureEncoded, CryptSignAndEncodeCertificate, CertDeleteCertificateFromStore, CryptFindOIDInfo, CertFindExtension, CryptVerifyCertificateSignature, CertDuplicateCertificateContext, CryptEncodeObject, CryptQueryObject, CertSaveStore, CertCreateCertificateContext, CryptImportPublicKeyInfo, CryptSignMessage, CryptExportPublicKeyInfo, CertFindCertificateInStore, CertControlStore, CertStrToNameW, CryptVerifyMessageSignature, CryptMsgUpdate, CryptDecodeObject, CertGetPublicKeyLength, CertGetNameStringW, CertVerifyCertificateChainPolicy, CertVerifyRevocation, CertGetEnhancedKeyUsage, CertEnumCRLsInStore, CertAddEncodedCertificateToStore, CertCompareCertificate, CertOpenStore, CertFreeCertificateContext, CertStrToNameA, CertOpenSystemStoreW, CryptVerifyMessageSignatureWithKey, CertSetCertificateContextProperty, CryptImportPublicKeyInfoEx2, CertGetIntendedKeyUsage, CertGetCertificateChain, CertAddStoreToCollection, CertVerifyTimeValidity, CryptUnprotectData, CertRemoveStoreFromCollection, CryptProtectMemory, CertCreateSelfSignCertificate, CryptUnprotectMemory, CryptProtectData, CertGetCertificateContextProperty, CertEnumCertificatesInStore, CryptXmlGetStatus, CryptXmlGetSignature, CryptXmlGetDocContext, CryptXmlGetReference, CryptXmlOpenToDecode, CryptXmlDigestReference, CryptXmlVerifySignature, DnsQueryConfig, FilterSendMessage, FilterConnectCommunicationPort, FilterGetMessage, CreateRectRgn, BitBlt, CreatePolygonRgn, CreateDIBSection, DeleteDC, CreateFontIndirectW, SelectObject, CreatePen, GetObjectW, CombineRgn, GetStockObject, StretchBlt, CreateDCW, FillRgn, DeleteObject, CreateRectRgnIndirect, RoundRect, GetDeviceCaps, SetRectRgn, CreateCompatibleDC, NotifyIpInterfaceChange, IcmpCreateFile, GetIpAddrTable, GetIpErrorString, GetAdaptersInfo, GetBestInterface, GetIfTable, NotifyRouteChange, IcmpSendEcho, GetNetworkParams, SendARP, CancelMibChangeNotify2, GetTcpTable, IcmpCloseHandle, GetAdaptersAddresses, InitializeCriticalSectionEx, GetStringTypeA, IsWow64Process, HeapReAlloc, GetModuleFileNameA, HeapSize, GetThreadId, DisconnectNamedPipe, VerifyVersionInfoA, GetStringTypeExW, CopyFileA, CompareStringA, GetLocaleInfoA, GetProfileStringW, IsDBCSLeadByte, SetNamedPipeHandleState, WinExec, GetThreadTimes, CreateFileMappingW, GetEnvironmentStrings, CreateProcessW, FileTimeToDosDateTime, RtlVirtualUnwind, GetFullPathNameA, SetEndOfFile, LCMapStringA, TlsAlloc, GetCurrentProcess, RegisterWaitForSingleObject, WritePrivateProfileSectionW, SetDllDirectoryW, GetDiskFreeSpaceA, SetProcessWorkingSetSize, SleepEx, Wow64EnableWow64FsRedirection, FindResourceExW, TryEnterCriticalSection, GetOEMCP, DeleteFileA, ReadFileEx, GetUserDefaultLCID, SetEnvironmentVariableA, CloseHandle, LockFile, GetVolumePathNamesForVolumeNameW, IsBadWritePtr, DefineDosDeviceW, GetProcessHeap, GetFileSizeEx, GetCommandLineW, lstrcatW, GetNumaHighestNodeNumber, GetProfileSectionW, FindResourceExA, TlsSetValue, GetProcessAffinityMask, CloseThreadpoolWait, GetSystemDirectoryW, MoveFileW, InitOnceExecuteOnce, LoadLibraryExA, WTSGetActiveConsoleSessionId, lstrcmpW, FreeEnvironmentStringsA, GetNumberOfConsoleInputEvents, FindFirstFileExW, LCMapStringW, AcquireSRWLockExclusive, SetFileAttributesW, CloseThreadpool, CreateTimerQueue, EnumSystemLocalesW, HeapValidate, TlsFree, AreFileApisANSI, ReleaseMutex, CreatePipe, SetFilePointerEx, SetProcessAffinityMask, DeleteFiber, QueryFullProcessImageNameA, GetDiskFreeSpaceExA, lstrcmpiA, WaitForThreadpoolTimerCallbacks, CompareStringW, VirtualLock, GetWindowsDirectoryA, DeleteTimerQueueEx, GetFinalPathNameByHandleW, CreateFiber, ExitProcess, GetStdHandle, FlushFileBuffers, PostQueuedCompletionStatus, GetTempFileNameA, FileTimeToLocalFileTime, GetPrivateProfileStringW, GetVersionExW, InterlockedFlushSList, VirtualFreeEx, LoadLibraryW, GetFileSize, GetEnvironmentStringsW, GetSystemWow64DirectoryW, FindFirstVolumeW, FlsFree, CreateThreadpoolTimer, FreeLibraryWhenCallbackReturns, VerSetConditionMask, ReleaseSRWLockShared, Sleep, FindFirstChangeNotificationW, QueueUserAPC, SetUnhandledExceptionFilter, OpenEventA, SetFileTime, GetSystemTimes, SetConsoleScreenBufferSize, OutputDebugStringA, SetFileAttributesA, GetProcessTimes, WaitNamedPipeA, QueryPerformanceCounter, FlsAlloc, TlsGetValue, WaitForSingleObject, EnumSystemLocalesA, SetHandleInformation, CloseThreadpoolCleanupGroup, GetModuleHandleExW, UnmapViewOfFile, GetShortPathNameW, OpenFile, SetHandleCount, LocalAlloc, LoadResource, GetModuleHandleExA, GetShortPathNameA, GetDriveTypeW, FlushInstructionCache, SetConsoleCursorPosition, lstrlenW, GetPrivateProfileStringA, DeleteCriticalSection, GetProcAddress, ExitThread, GetQueuedCompletionStatus, GetTickCount64, GlobalUnlock, DosDateTimeToFileTime, ConvertFiberToThread, TryAcquireSRWLockExclusive, GetPrivateProfileSectionW, CreateDirectoryW, GetLogicalProcessorInformation, OpenFileMappingA, DeviceIoControl, SystemTimeToTzSpecificLocalTime, TryAcquireSRWLockShared, CancelWaitableTimer, IsProcessorFeaturePresent, SetThreadpoolThreadMaximum, RemoveDirectoryW, GetOverlappedResult, TerminateThread, ReleaseSRWLockExclusive, Wow64RevertWow64FsRedirection, GetTimeZoneInformation, RtlPcToFileHeader, SetWaitableTimer, GetStartupInfoW, MulDiv, SetThreadpoolThreadMinimum, WaitForMultipleObjects, SetThreadLocale, GetVolumeInformationW, CreateDirectoryA, InterlockedExchange, CreateThread, lstrcpyW, SetDllDirectoryA, WaitNamedPipeW, InterlockedCompareExchange, FindNextChangeNotification, CloseThreadpoolCleanupGroupMembers, RtlUnwind, CreateThreadpool, GetLocaleInfoEx, InitializeCriticalSection, WriteProcessMemory, ConnectNamedPipe, CompareFileTime, LoadLibraryExW, CreateMailslotW, GetTempPathA, GetVolumeInformationByHandleW, FindCloseChangeNotification, VirtualAlloc, GetDiskFreeSpaceW, CreateEventExW, InterlockedPushEntrySList, IsDebuggerPresent, GetVersionExA, GlobalMemoryStatus, GetFirmwareEnvironmentVariableW, HeapDestroy, GetLogicalDriveStringsA, QueueUserWorkItem, GetCPInfo, GetSystemPowerStatus, SuspendThread, GetEnvironmentVariableW, WaitForMultipleObjectsEx, RtlUnwindEx, GetFileAttributesExA, GetThreadContext, WideCharToMultiByte, VirtualProtectEx, CreateWaitableTimerW, GetLogicalDriveStringsW, GetTempPathW, GetSystemInfo, FileTimeToSystemTime, UnlockFile, CopyFileW, FindVolumeClose, SetThreadpoolTimer, WriteProfileStringW, FindFirstFileW, UnregisterWait, UnlockFileEx, RtlLookupFunctionEntry, Process32NextW, GetProcessId, GlobalMemoryStatusEx, AddAtomW, WaitForSingleObjectEx, GetEnvironmentVariableA, GetFullPathNameW, TerminateProcess, SystemTimeToFileTime, CreateFileA, GetCurrentThread, GetTimeFormatA, InterlockedPopEntrySList, GetTickCount, GetDateFormatA, FindNextFileW, CheckRemoteDebuggerPresent, GetFileAttributesExW, WritePrivateProfileStringW, CreateFileMappingA, QueryUnbiasedInterruptTime, ExpandEnvironmentStringsA, GetModuleFileNameW, SetStdHandle, RemoveDirectoryA, GetComputerNameExW, GetLogicalProcessorInformationEx, CreateThreadpoolWork, QueryFullProcessImageNameW, CreateMutexW, DeleteFileW, CancelIo, FindFirstChangeNotificationA, UnhandledExceptionFilter, GetFileType, UnregisterWaitEx, SetThreadContext, VirtualQuery, CreateSemaphoreA, OpenEventW, GetDateFormatW, QueryDosDeviceW, WriteFile, GetMailslotInfo, CreateToolhelp32Snapshot, FlushViewOfFile, GetDynamicTimeZoneInformation, IsDBCSLeadByteEx, CreateDirectoryExW, GetFileAttributesW, GetDriveTypeA, lstrcpynA, InitializeSRWLock, DebugBreak, FindResourceW, QueryDepthSList, OpenThread, SignalObjectAndWait, GetFileAttributesA, MapViewOfFile, GetModuleHandleW, EndUpdateResourceW, DeleteTimerQueue, SetConsoleCtrlHandler, UpdateResourceW, MoveFileExW, HeapCompact, GetConsoleCP, FlushConsoleInputBuffer, CreateProcessA, GetPrivateProfileSectionNamesA, RaiseException, OutputDebugStringW, FindFirstFileA, GetFileInformationByHandle, PeekNamedPipe, PulseEvent, SetThreadUILanguage, CreateEventA, GetLocalTime, GetSystemDefaultLangID, GetConsoleOutputCP, GetPrivateProfileSectionA, CreateMutexA, FreeLibraryAndExitThread, InitOnceComplete, FlsSetValue, ReleaseSemaphore, SetConsoleTextAttribute, LockResource, GetConsoleMode, GetTempFileNameW, GetExitCodeThread, SetPriorityClass, RtlCaptureContext, SetLastError, GetSystemWindowsDirectoryW, FindResourceA, InitializeConditionVariable, GetCommandLineA, VirtualUnlock, lstrcpyA, VirtualFree, InterlockedDecrement, WakeAllConditionVariable, GetNativeSystemInfo, VirtualQueryEx, CreateNamedPipeW, SleepConditionVariableCS, GetComputerNameA, ReadProcessMemory, OpenFileMappingW, lstrcpynW, IsValidCodePage, DisableThreadLibraryCalls, OpenProcess, WakeConditionVariable, GlobalReAlloc, RemoveVectoredExceptionHandler, FatalAppExitA, GetComputerNameW, GetPriorityClass, ReadFile, GetSystemTime, SwitchToFiber, GetLongPathNameW, SetThreadAffinityMask, LCMapStringEx, WriteProfileSectionW, IsBadReadPtr, SubmitThreadpoolWork, lstrcmpiW, InterlockedIncrement, CloseThreadpoolTimer, FindFirstFileExA, BeginUpdateResourceW, GetACP, GetThreadLocale, GetProcessWorkingSetSize, HeapAlloc, ResumeThread, GetConsoleScreenBufferInfo, ExpandEnvironmentStringsW, HeapCreate, VerifyVersionInfoW, SwitchToThread, GetLastError, CreateEventW, DeleteTimerQueueTimer, CreateThreadpoolWait, FlushProcessWriteBuffers, GlobalFree, GetFileTime, FormatMessageW, FreeLibrary, SleepConditionVariableSRW, CreateIoCompletionPort, SetErrorMode, GetTimeFormatW, GetExitCodeProcess, WaitForThreadpoolWorkCallbacks, GetComputerNameExA, SetThreadpoolWait, GlobalAlloc, CopyFileExW, SetEvent, BindIoCompletionCallback, CreateSemaphoreW, CreateSymbolicLinkW, MultiByteToWideChar, MoveFileA, VirtualProtect, GetUserDefaultLangID, CloseThreadpoolWork, GetUserDefaultUILanguage, GetSystemDefaultLCID, FindNextFileA, GetPrivateProfileSectionNamesW, CreateHardLinkW, GetFileInformationByHandleEx, GetLogicalDrives, GlobalSize, VirtualAllocEx, LocalFree, SetEnvironmentVariableW, ChangeTimerQueueTimer, GetStringTypeW, HeapSetInformation, CreateThreadpoolCleanupGroup, MoveFileExA, FreeConsole, InitializeSListHead, GetModuleHandleA, GetDiskFreeSpaceExW, AddVectoredExceptionHandler, CreateTimerQueueTimer, lstrcmpA, GetPrivateProfileIntA, CancelIoEx, LockFileEx, LocalFileTimeToFileTime, GetSystemTimeAsFileTime, SetThreadPriority, FindNextVolumeW, SetFilePointer, CompareStringEx, InitializeCriticalSectionAndSpinCount, ConvertThreadToFiber, RtlCaptureStackBackTrace, AcquireSRWLockShared, FindClose, GetWindowsDirectoryW, GlobalLock, SizeofResource, ResetEvent, InitOnceBeginInitialize, FreeEnvironmentStringsW, DuplicateHandle, SetFileInformationByHandle, GetLocaleInfoW, MapViewOfFileEx, lstrlenA, GetSystemDirectoryA, LoadLibraryA, GetCurrentProcessId, CreateSemaphoreExW, GetThreadSelectorEntry, LeaveCriticalSection, Process32FirstW, GetCurrentProcessorNumber, GetVersion, EnterCriticalSection, GetThreadPriority, IsValidLocale, HeapFree, GetLongPathNameA, QueryPerformanceFrequency, CreateFileW, GetCurrentThreadId, SetConsoleMode, FlsGetValue, IsBadStringPtrA, GetVolumeInformationA, DeleteAtom, FormatMessageA, Wow64DisableWow64FsRedirection, WNetAddConnection2W, WNetGetConnectionA, WNetCancelConnection2A, WNetEnumResourceW, WNetGetConnectionW, WNetCancelConnection2W, WNetAddConnection2A, WNetOpenEnumW, WNetGetLastErrorW, WNetCloseEnum, NetUserDel, NetServerGetInfo, NetShareDel, NetShareAdd, NetRemoteTOD, NetShareGetInfo, NetGetDCName, NetWkstaGetInfo, NetShareEnum, NetGetAnyDCName, DsRoleGetPrimaryDomainInformation, NetUserAdd, NetUserEnum, DsGetDcNameW, NetApiBufferFree, DsGetDcNameA, DsRoleFreeMemory, NetUserGetInfo, NetScheduleJobAdd, IdnToUnicode, IdnToAscii, SafeArrayUnaccessData, SafeArrayCopy, SysAllocStringLen, SysStringLen, BSTR_UserSize, SetErrorInfo, VarUI4FromStr, SafeArrayGetElement, VariantCopyInd, LoadTypeLib, RegisterTypeLib, VarCmp, SafeArrayDestroy, SystemTimeToVariantTime, SafeArrayGetDim, LPSAFEARRAY_UserUnmarshal, SafeArrayCreateVector, SafeArrayRedim, SysAllocStringByteLen, SysStringByteLen, BSTR_UserUnmarshal, SafeArrayUnlock, VarBstrCmp, SafeArrayGetLBound, LPSAFEARRAY_UserSize, LPSAFEARRAY_UserFree, LPSAFEARRAY_UserMarshal, SysAllocString, VariantCopy, CreateErrorInfo, SafeArrayCreate, VariantChangeType, GetErrorInfo, SafeArrayPutElement, VariantClear, SafeArrayLock, SafeArrayDestroyData, SysFreeString, VariantInit, SafeArrayAccessData, LoadRegTypeLib, SafeArrayGetVartype, BSTR_UserMarshal, SafeArrayGetUBound, VariantTimeToSystemTime, UnRegisterTypeLib, BSTR_UserFree, GetPwrCapabilities, SetSuspendState, EnumProcesses, GetModuleBaseNameA, GetProcessImageFileNameA, GetModuleFileNameExW, RpcServerInqBindings, RpcServerUnregisterIfEx, RpcStringBindingComposeA, NdrAsyncServerCall, CStdStubBuffer_AddRef, NdrClientCall2, RpcEpRegisterNoReplaceA, NdrOleAllocate, I_RpcBindingInqLocalClientPID, RpcMgmtIsServerListening, DceErrorInqTextW, RpcStringBindingParseW, RpcServerUseProtseqEpA, NdrDllGetClassObject, RpcServerUseProtseqEpW, RpcStringBindingComposeW, NdrOleFree, RpcServerInqCallAttributesW, NdrClientCall3, RpcAsyncCompleteCall, IUnknown_AddRef_Proxy, UuidToStringA, IUnknown_QueryInterface_Proxy, CStdStubBuffer_Invoke, CStdStubBuffer_QueryInterface, RpcAsyncInitializeHandle, UuidCreate, RpcServerRegisterIfEx, RpcStringFreeW, CStdStubBuffer_DebugServerQueryInterface, RpcBindingFree, RpcStringFreeA, CStdStubBuffer_IsIIDSupported, RpcServerListen, NdrAsyncClientCall, RpcEpUnregister, RpcAsyncCancelCall, CStdStubBuffer_CountRefs, RpcBindingFromStringBindingA, RpcImpersonateClient, RpcBindingSetAuthInfoW, CStdStubBuffer_Disconnect, RpcBindingVectorFree, UuidToStringW, NdrDllRegisterProxy, NdrDllUnregisterProxy, RpcMgmtStopServerListening, IUnknown_Release_Proxy, NdrStubForwardingFunction, RpcBindingFromStringBindingW, UuidFromStringW, RpcRevertToSelfEx, CStdStubBuffer_DebugServerRelease, RpcRaiseException, RpcServerRegisterIf2, RpcServerInqCallAttributesA, NdrServerCall2, CStdStubBuffer_Connect, RpcMgmtWaitServerListen, RpcServerUnregisterIf, NdrDllCanUnloadNow, NdrCStdStubBuffer2_Release, RpcBindingToStringBindingW, NdrCStdStubBuffer_Release, SetupFindNextMatchLineW, CM_Set_DevNode_Problem_Ex, SetupDiGetDeviceInstallParamsW, SetupOpenLog, SetupGetStringFieldW, SetupIterateCabinetW, SetupDiGetClassDevsExA, SetupDiEnumDeviceInfo, SetupGetLineCountW, SetupLogErrorW, SetupDiGetActualSectionToInstallW, SetupDiSetDeviceInstallParamsW, SetupDiGetSelectedDriverW, SetupOpenInfFileW, SetupPromptReboot, SetupFindFirstLineW, SetupDiDestroyDeviceInfoList, SetupDiGetDeviceRegistryPropertyW, SetupCloseInfFile, SetupCloseLog, SetupDiGetDriverInfoDetailW, ShellExecuteExW, SHAssocEnumHandlers, SHGetFolderPathW, SHGetPropertyStoreFromParsingName, CommandLineToArgvW, SHFileOperationW, SHGetKnownFolderPath, SHGetFolderPathAndSubDirW, SHGetFolderPathA, ShellExecuteW, DragQueryFileW, DragAcceptFiles, SHGetSpecialFolderPathW, PathIsUNCServerW, PathFileExistsA, PathAddBackslashW, PathIsUNCServerShareW, PathRemoveBackslashW, StrStrIW, PathCanonicalizeW, PathFindFileNameW, PathFileExistsW, PathRemoveExtensionW, PathIsUNCW, PathIsDirectoryW, PathIsRelativeW, PathAppendW, PathRenameExtensionW, PathStripToRootW, PathCombineW, SHDeleteKeyW, PathRemoveFileSpecW, PathAddExtensionW, PathFindExtensionW, PathStripPathW, FreeCredentialsHandle, LsaLookupAuthenticationPackage, QueryCredentialsAttributesA, LsaFreeReturnBuffer, GetUserNameExW, DeleteSecurityContext, LsaGetLogonSessionData, InitializeSecurityContextA, LsaConnectUntrusted, LsaEnumerateLogonSessions, FreeContextBuffer, AcquireCredentialsHandleA, LsaCallAuthenticationPackage, PostThreadMessageW, MonitorFromWindow, MapWindowPoints, CreateWindowExW, CharLowerBuffW, SetMenuItemBitmaps, IsCharAlphaW, KillTimer, DestroyWindow, SetTimer, DdeClientTransaction, EndDialog, ShowWindow, DdeDisconnect, EnumDisplayMonitors, wsprintfA, DispatchMessageW, OffsetRect, PostQuitMessage, DdeConnect, GetDlgItem, LoadBitmapW, InvalidateRect, DdeInitializeA, DdeCreateStringHandleA, DdeUninitialize, MessageBoxA, GetWindowTextW, EnumWindows, SetRectEmpty, SystemParametersInfoW, GetWindowDC, UnregisterClassW, GetClassLongW, SendMessageA, ScreenToClient, CharToOemBuffA, TranslateMessage, GetClassNameW, GetUserObjectInformationA, SetCursor, LockWindowUpdate, GetProcessWindowStation, SetWindowLongW, GetWindowRect, SetWindowLongPtrW, SetClipboardData, DdeFreeStringHandle, GetDesktopWindow, CharNextW, PeekMessageW, ExitWindowsEx, GetMonitorInfoW, RegisterClassExW, CreateDialogParamW, CharPrevExA, SetLayeredWindowAttributes, MoveWindow, BeginPaint, IsWindowVisible, GetWindowLongW, CharNextA, InflateRect, RegisterWindowMessageA, CharLowerW, RedrawWindow, LoadStringA, UnregisterClassA, EndPaint, CloseClipboard, OpenClipboard, LoadStringW, UpdateWindow, IsDialogMessageW, GetWindowTextLengthW, InsertMenuW, GetWindowThreadProcessId, GetClassInfoA, CharUpperA, DefWindowProcW, IsWindowEnabled, GetClientRect, SetRect, CharUpperW, DdeCreateDataHandle, LoadCursorW, DisableProcessWindowsGhosting, LoadAcceleratorsW, GetParent, DialogBoxParamW, GetMessageW, GetWindowPlacement, SetWindowPos, SetWindowTextW, GetSysColorBrush, GetWindowLongPtrW, MessageBoxW, GetUserObjectInformationW, IsCharAlphaNumericW, EmptyClipboard, TranslateAcceleratorW, DrawIconEx, DestroyIcon, LoadImageW, EnableWindow, LoadIconW, PostMessageW, wsprintfW, SetDlgItemTextW, GetSystemMetrics, SendMessageW, GetWindow, WaitForInputIdle, LoadUserProfileW, UnloadUserProfile, GetProfileType, ExpandEnvironmentStringsForUserW, ExpandEnvironmentStringsForUserA, CreateEnvironmentBlock, DestroyEnvironmentBlock, WinHttpGetDefaultProxyConfiguration, WinHttpOpen, WinHttpAddRequestHeaders, WinHttpQueryHeaders, WinHttpCloseHandle, WinHttpSetStatusCallback, WinHttpReadData, WinHttpOpenRequest, WinHttpReceiveResponse, WinHttpQueryAuthSchemes, WinHttpConnect, WinHttpSetCredentials, WinHttpCrackUrl, WinHttpGetProxyForUrl, WinHttpSendRequest, WinHttpQueryDataAvailable, WinHttpGetIEProxyConfigForCurrentUser, HttpSendRequestW, InternetSetOptionW, InternetQueryOptionW, InternetConnectW, InternetSetStatusCallbackW, HttpQueryInfoW, InternetCloseHandle, InternetCrackUrlW, InternetOpenW, HttpOpenRequestW, InternetReadFile, CryptCATAdminAcquireContext2, CryptCATAdminEnumCatalogFromHash, WTHelperProvDataFromStateData, CryptCATAdminCalcHashFromFileHandle2, CryptCATAdminAddCatalog, WTHelperGetProvSignerFromChain, WTHelperGetProvCertFromChain, CryptCATAdminAcquireContext, WinVerifyTrust, CryptCATCatalogInfoFromContext, CryptCATAdminReleaseContext, CryptCATAdminReleaseCatalogContext, CryptCATAdminCalcHashFromFileHandle, __WSAFDIsSet, WSASocketW, bind, inet_ntop, setsockopt, sendto, send, recv, WSAGetLastError, gethostbyaddr, WSAWaitForMultipleEvents, closesocket, gethostbyname, WSAGetOverlappedResult, WSAEventSelect, WSAStartup, listen, inet_pton, ntohl, WSAResetEvent, ntohs, recvfrom, WSAPoll, WSASetLastError, WSACloseEvent, shutdown, WSARecv, InetNtopW, WSACleanup, WSAStringToAddressW, getpeername, WSACreateEvent, getnameinfo, inet_ntoa, WSASend, WSAEnumNetworkEvents, select, InetPtonW, gethostname, htons, FreeAddrInfoW, WSASetEvent, freeaddrinfo, getsockname, getsockopt, socket, GetAddrInfoW, getaddrinfo, inet_addr, htonl, WSAIoctl, accept, connect, WSAAddressToStringW, ioctlsocket, getprotobynumber, getservbyname, getprotobyname, getservbyport, WTSEnumerateSessionsW, WTSQuerySessionInformationW, WTSFreeMemory, WTSQueryUserToken, RoOriginateError, _getwch, mbtowc, _wtol, wcstod, atof, strtof, atol, _strtoui64, wcstoll, wcstoul, wcstombs, _atoi64, _wtoll, _wcstoi64, _wtoi, _ltoa_s, _i64toa, strtoul, _i64tow_s, strtol, _ultoa_s, _itoa, mbstowcs, strtoll, wcstoull, btowc, strtod, _ultoa, _itoa_s, _wcstoui64, _strtoi64, wctomb_s, atoi, _ui64tow_s, _wtoi64, wcstol, strtoull, wcstombs_s, _wgetcwd, _putenv_s, _putenv, getenv, _wgetenv, _unlink, rename, _chdir, _fstat32, _umask, _fstat64i32, _getdrive, _chdrive, _splitpath_s, _access, _findnext32, _stat64, _stat32i64, _waccess, _wstat64i32, _wrmdir, _makepath_s, _wunlink, _mkdir, _findfirst32, remove, _wremove, _wmkdir, _wchdir, _chmod, _findclose, _lock_file, _wsplitpath_s, _wmakepath_s, _stat32, _wrename, _wfullpath, _wstat64, _fstat64, _stat64i32, _getdrives, _rmdir, _unlock_file, _set_new_mode, realloc, _callnewh, calloc, free, _recalloc, malloc, localeconv, _configthreadlocale, setlocale, _isnan, exp, frexp, _fdopen, round, nextafter, log2, ceil, ldexp, copysign, sqrt, _mbsnbcpy_s, _ismbblead, _mbscmp, _mbsstr, _mbsicmp, memmove, memcpy, wcsstr, wcschr, _CxxThrowException, wcsrchr, memcmp, _spawnve, _spawnv, _cexit, _exit, _wassert, _beginthread, _set_invalid_parameter_handler, _resetstkoflw, _invalid_parameter_noinfo, abort, _set_abort_behavior, _endthreadex, _c_exit, _invalid_parameter_noinfo_noreturn, raise, strerror, _controlfp_s, _beginthreadex, _set_new_handler, _initterm, _initterm_e, _getpid, perror, strerror_s, exit, _seh_filter_exe, signal, _lseek, _getcwd, _open, fputs, _wfopen_s, freopen, _lseeki64, fputc, _filelength, _open_osfhandle, _fileno, putc, fputwc, fread_s, ungetc, putchar, _ftelli64, setbuf, _close, getc, _sopen_s, fgetpos, fseek, setvbuf, ungetwc, _dup, rewind, _pclose, fsetpos, _fseeki64, puts, fopen, fgets, fgetc, _set_fmode, _wfopen, _putws, _dup2, _read, _setmode, _wopen, fclose, ftell, _mktemp, fopen_s, _pipe, _get_osfhandle, _fsopen, fwrite, fgetwc, tmpnam, ferror, feof, _popen, _write, fread, fgetws, fflush, clearerr, _wfsopen, strcpy_s, _wcsupr_s, strtok, _wcsdup, isalnum, wcspbrk, strncmp, wcscpy, wcscat_s, ispunct, iswalpha, isupper, wcsncat, towlower, wcsncmp, towupper, isalpha, _strdup, strcspn, _wcsicmp, strpbrk, toupper, wcscmp, _strupr, strcat_s, wcscspn, strcpy, wcscat, iswpunct, islower, _strnicmp, strncpy_s, wcsncat_s, strlen, wcsspn, wcstok, wcsncpy, iswprint, strncat, wcsncpy_s, strncat_s, _strupr_s, iswdigit, isxdigit, _wcslwr, wcslen, strspn, isdigit, wcsnlen, strcmp, _strlwr_s, strcat, isspace, strtok_s, wmemcpy_s, wcscpy_s, iswctype, _stricmp, iscntrl, wcstok_s, tolower, isgraph, _wcsupr, _wcsnicmp, isprint, strnlen, iswspace, mblen, _wcslwr_s, memset, strncpy, _gmtime64, _gmtime32, _difftime64, _ftime32, _ftime64_s, _ctime64, _wctime64_s, _tzset, _gmtime64_s, _mkgmtime32, _strtime, _mkgmtime64, _time64, strftime, asctime, _difftime32, _strdate, _utime32, wcsftime, _localtime32, _time32, _localtime64, _mktime64, _ctime64_s, _mktime32, _ctime32, _localtime64_s, clock, labs, qsort, ldiv, srand, rand, rand_s, bsearch, BCryptImportKey, BCryptFinishHash, BCryptDecrypt, BCryptExportKey, BCryptDeriveKey, BCryptOpenAlgorithmProvider, BCryptVerifySignature, BCryptSecretAgreement, BCryptEncrypt, BCryptCloseAlgorithmProvider, BCryptDestroyKey, BCryptImportKeyPair, BCryptGenRandom, BCryptHashData, BCryptGetProperty, BCryptFinalizeKeyPair, BCryptCreateHash, BCryptSignHash, BCryptDestroyHash, BCryptDestroySecret, BCryptGenerateKeyPair, BCryptSetProperty, BCryptHash, CredUIParseUserNameW, D3D11CreateDevice, SymGetModuleBase, SymInitialize, SymFromAddr, SymCleanup, CreateDXGIFactory1, MapFileAndCheckSumW, ZwSetEvent, _snprintf, RtlAreBitsSet, NtOpenKey, ZwQueryInformationFile, NtQueryInformationProcess, RtlGetDaclSecurityDescriptor, NtWaitForSingleObject, RtlFindClearBitsAndSet, ZwOpenSection, RtlUnicodeToUTF8N, RtlInitUnicodeString, NtCreateFile, RtlDowncaseUnicodeChar, NtQuerySystemInformation, _itow, NtAllocateVirtualMemory, ZwCreateFile, RtlGetElementGenericTable, NtOpenEvent, RtlInitializeGenericTable, ZwQueryInformationToken, RtlLookupElementGenericTable, NtOpenProcess, ZwWriteFile, RtlNtStatusToDosError, ZwDuplicateObject, RtlCreateSecurityDescriptor, NtClose, NtWriteFile, strrchr, ZwQueryObject, RtlIpv6AddressToStringW, ZwOpenDirectoryObject, RtlAddAccessAllowedAce, NtQueryValueKey, NtQueryVirtualMemory, ZwQueryFullAttributesFile, ZwQuerySecurityObject, ZwOpenEvent, NtQueryInformationToken, ZwDeviceIoControlFile, RtlEqualUnicodeString, ZwWaitForSingleObject, ZwQuerySymbolicLinkObject, RtlFreeHeap, memchr, ZwAllocateVirtualMemory, RtlAllocateHeap, NtFreeVirtualMemory, RtlIsGenericTableEmpty, ZwCreateSection, RtlCreateHeap, NtSetInformationThread, _memicmp, ZwOpenKey, RtlSetDaclSecurityDescriptor, ZwTerminateProcess, _vsnwprintf, ZwSetSecurityObject, RtlTimeToTimeFields, swprintf_s, RtlCompareUnicodeString, RtlFreeUnicodeString, ZwOpenSymbolicLinkObject, RtlIpv4AddressToStringW, RtlInsertElementGenericTable, RtlGetVersion, _vsnprintf, NtQueryInformationThread, RtlDestroyHeap, RtlInitAnsiString, RtlConvertSidToUnicodeString, NtQueryObject, strchr, NtDuplicateObject, strstr, NtOpenProcessToken, NtCreateEvent, ZwFreeVirtualMemory, RtlClearBits, RtlCreateAcl, ZwUnmapViewOfSection, sscanf, RtlEqualSid, RtlGUIDFromString, RtlDeleteElementGenericTable, NtSetEvent, ZwSetInformationFile, ZwCreateEvent, ZwOpenFile, ZwMapViewOfSection, ZwReadFile, sprintf, ZwSetInformationThread, ZwQueryVirtualMemory, ZwClose, CreateILockBytesOnHGlobal, ReleaseStgMedium, CreateStreamOnHGlobal, CoTaskMemFree, StgCreateDocfileOnILockBytes, CoGetClassObject, CoCreateGuid, CoCreateFreeThreadedMarshaler, CoAddRefServerProcess, CoUninitialize, CoMarshalInterface, OleRun, CoTaskMemRealloc, CLSIDFromString, CoUnmarshalInterface, CoLockObjectExternal, CoRevokeClassObject, CoTaskMemAlloc, CoRegisterClassObject, CoSetProxyBlanket, CoInitialize, CoInitializeSecurity, StringFromCLSID, CoResumeClassObjects, CoMarshalInterThreadInterfaceInStream, CoReleaseServerProcess, CoCreateInstance, CLSIDFromProgID, CoGetInterfaceAndReleaseStream, CoInitializeEx, CoReleaseMarshalData, CoIsHandlerConnected, CoFreeUnusedLibraries, StringFromGUID2, PdhEnumObjectItemsW, PdhOpenQueryW, PdhGetFormattedCounterValue, PdhAddCounterW, PdhCollectQueryData, PdhAddEnglishCounterA, PdhCloseQuery, TdhGetEventInformation, TdhGetPropertySize, EvtQuery, EvtSubscribe, EvtClose, EvtOpenSession, EvtNext, EvtCreateRenderContext and EvtRender.

The TOE does not request memory mapping to any explicit address. The TOE has protection mechanisms to prevent attacks during its execution. To perform this, a continuous scan is performed to all files that are used or opened with the aim of detecting unusual activity and operating system process that attempt to attack the TOE. The TOE also implements a group of protection mechanisms when compiling occurs. To achieve this, the TOE includes the following compilation flags: ASLR, DEP and GS. On a specific basis, the following compiler flags have been used to enable ASLR when the application is compiled:

- */HIGHENTROPYVA* which specifies whether the executable image supports high-entropy 64-bit address space layout randomization (ASLR).
- */DYNAMICBASE,* which specifies whether to generate an executable image that can be randomly rebased at load time by using the address space layout randomization (ASLR) feature of Windows Operating Systems.

This is addressed by **FPT_AEX_EXT.1**. Moreover, the TOE includes a *Self-Protection* component designed to provide protection to 3rd party software used by the TOE itself. This protection is also useful if an attacker can disable all ASLR and DEP protection or perform local privilege escalation. This protection uses kernel callback functions to monitor the access to the TOE and disallow injection and other type of attacks.

Regarding the software updates, the TOE allows to check the installed version providing this information on the main interface of the TOE (at the bottom right). This version is performed by the Check Point internal versioning control (according to **FPT_IDV_EXT.1**). The number version of the TOE is composed of major version, minor version and build number version separated by points. Moreover, the TOE allows checking the updates manually with the aim of installing new available updates. It is necessary to use the "Update now" option, where it acts as updating the product and proceeds to apply the available updates if necessary. The TOE also has a mechanism that allows installing software updates when available. This mechanism allows to install automatically the available updates from the web interface of the Harmony Endpoint EPMaaS and its corresponding execution on clients where the TOE is installed. In this way, the TOE takes the necessary steps to verify the integrity of the update packages installed. This verification is carried out using SHA-256 hash algorithm in order to check the integrity of the update package. It is necessary to take into account that the update package shall always come from an authorized source as the Check Point official repository (Check Point Software Technologies Ltd.). The TOE is distributed through Harmony Endpoint EPMaaS. The updates are automatically displayed in Harmony Endpoint EPMaaS and, when it communicates with the TOE, verifies whether the version is different and proceeds to deploy the update. The TOE does not download, modify, replace or update its own binary code. The algorithm used to perform the signature of installer and updates is RSA-PKCS1v1.5 with SHA256 using an RSA 4096 bits key and is countersigned by DigiCert in order to establish a valid date by applying timestamp. This is addressed by **FPT_TUD_EXT.1**.

The TOE only uses the list of third-party libraries included in the Annex1 document attached in conjunction with the current Security Target to prevent the use of components that could present a privacy threat and ensure that technical vulnerabilities are correctly addressed. This is addressed by **FPT_LIB_EXT.1**.

The TOE is always distributed in *.exe* format and its distribution can be performed by means of the Harmony Endpoint EPMaaS automatically or manually. The TOE executable is packaged in a specific way that, if removed, leaves no trace of its installation (except for configurations and output or audit files). As mentioned above, virus database signatures and product updates are signed (by using the RSA-PKCS1v1.5 algorithm with SHA256) and these updates always come from an authorized source (Check Point Software Technologies Ltd.). This is addressed by **FPT_TUD_EXT.2**.

7.1.5 Trusted Path/Channels

The TOE establishes two secure communication channels by using different mechanisms to each channel:

- One channel consists of communication is implemented by the product and uses the TLS1.2 protocol and validating the certificates against the ones in the certificate store of the Windows operating system.
- Another channel consists of communications relying on operating system functionalities, making use of the HTTPS and TLS protocols and the cipher suites allowed by Windows and using the certificate store of the Windows operating system for validation.

Both channels are used for connections with the servers.

According to the first communication channel, the TOE encrypts all transmitted sensitive data with HTTPS and TLS protocols. These protocols protect the data during its transmission between the product and the external servers. The TOE only supports the TLSv1.2 communication protocol. Therefore, the transmitted data is encrypted via HTTPS and TLSv1.2 during the TOE's operation. The TOE allows the use of the HTTPS protocol to carry out the communication with the external servers. This is addressed by **FTP_DIT_EXT.1**. The use of this communication channel allows a secure transmission of the following type of data:

- Policy downloads and configurations.
- TOE updates.
- New virus database download.
- Heartbeat (a periodic client connection to the server).
- Application Control queries.
- Log uploads.

7.2 Timely Security Updates

The product is periodically updated and users are notified through the *Alert* section of its support center. This notifies the user of any discovered TOE-related vulnerability and the necessary steps to solve it. Similarly, if the solution consists of updating the product, it provides the necessary steps. Depending on the vulnerability's severity, the corresponding update will be issued with greater or lesser priority.

The TOE allows to check the installed version and also allows checking the updates manually with the aim of installing new available updates.

The TOE is updated based on the comparison of its version against the version of the deployed policy. Therefore, when a product updates its policy and detects that the version installed is different from the one indicated in the policy, it starts downloading and installing the new version. When a new update is available, a window will be displayed indicating this event and providing the option to install or postpone the update (the *Postpone* option is not always available).

The updates are provided as *exe* files on a monthly basis. They are signed by Check Point SHA256 and countersigned by DigiCert to establish a valid date by applying timestamp.

Check Point is committed to the security of its products. The security response team in Check Point is dedicated to respond to potential security problems and making sure reports on such issues are handled properly.

Check Point provides multiple ways to communicate with the security response team:

- Contact the security response team via the security-alert@checkpoint.com mailing list. It is necessary to include as many details as possible. It is possible to use the Check Point PGP key to encrypt the communication.
- Fill in the form included in the *Report a Potential Security Issue* webpage. Pressing "*Submit*" will send the report to the security response team.
- Contact technical services and mention the necessity to report a security issue. There is no need for a support contract in order to submit such report. The technical services personnel will make sure the report is escalated to the security response team.

Check Point provides an additional Incident Response service intended to be used in an Emergency. This service is a proven 24x7x365 security incident handling service. Check Point has a dedicated expert team to respond immediately to any security attack and accelerate the threat resolution. It is not necessary to be a current client or Check Point customer for this initial contact. On average, a vulnerability is resolved in between 9 hours and 30 days.

This information addresses the **ALC_TSU_EXT** security assurance requirement.

8 Acronyms

The following table shows the acronyms used in this document.

Acronym	Meaning
PP	Protection Profile
CC	Common Criteria
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFi	TSF Interface
OSP	Organisational Security Policies
EAL	Evaluation Assurance Level
ST	Security Target
IT	Information Technology
PII	Personally Identifiable Information
EPMaaS	Endpoint Management as a Service
SAN	Subject Alternative Name
CA	Certificate Authority

Table 8 *Abbreviations*

9 Glossary of Terms

Term	Meaning
PP-Module	An implementation-independent statement of security needs for a TOE type complementary to one or more Base Protection Profiles.
TOE Security Functionality	The security functionality of the product under evaluation.
TOE Summary Specification	A description of how a TOE satisfies the SFRs in a ST.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Assurance Requirement (SAR)	A requirement to assure the security of the TOE.
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.
Mobile Code	Software transmitted from a remote system for execution within a limited execution environment on the local system. Typically, there is no persistent installation and execution begins without the user's consent or even notification. Examples of mobile code technologies include JavaScript, Java applets, Adobe Flash, and Microsoft Silverlight.
Operating System (OS)	Software that manages hardware resources and provides services for applications.

Term	Meaning
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. [OMB]
Platform	The environment in which application software runs. The platform can be an operating system, hardware environment, a software based execution environment, or some combination of these. These types platforms may also run atop other platforms.
Sensitive Data	Sensitive data may include all user or enterprise data or may be specific application data such as emails, messaging, documents, calendar items, and contacts. Sensitive data must minimally include PII, credentials, and keys. Sensitive data shall be identified in the application's TSS by the ST author.
Stack Cookie	An anti-exploitation feature that places a value on the stack at the start of a function call, and checks that the value is the same at the end of the function call. This is also referred to as Stack Guard, or Stack Canaries.
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.
Augmentation	Addition of one or more requirement(s) to a package
Evaluation Assurance Level	Set of assurance requirements drawn from CC Part 3, representing a point on the CC predefined assurance scale, that form an assurance package
Operational Environment	Environment in which the TOE is operated
Protection Profile	Implementation-independent statement of security needs for a TOE type
Security Target	Implementation-dependent statement of security needs for a specific identified TOE
Target Of Evaluation	Set of software, firmware and/or hardware possibly accompanied by guidance

Table 9 Glossary of terms

10 Document References

The following table shows the documentation referenced in this document.

Reference	Document
[CC31R5P1]	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, Part 1: Introduction and general model
[CC31R5P2]	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, Part 2: Security functional components
[CC31R5P3]	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, Part 3: Security assurance components
[CEM31R5P3]	Common Criteria Evaluation methodology, Version 3.1, Revision 5
[SP2788]	Non-proprietary Security Policy Check Point CryptoCore version 4.0 FIPS 140-2
[Annex1]	Annex with the third-party libraries used by the TOE, Version 1.2.
[PPAPP14]	NIAP Protection Profile for Application Software version 1.4, 07 October 2021.
[PKGTLS11]	NIAP Functional Package for TLS Version 1.1, dated 12 February 2019.

Table 10 List of document references