Palo Alto Networks GlobalProtect App Version 5.1.5 Security Target

Version 1.0
May 8, 2020

Palo Alto Networks, Inc.
3000 Tannery Way
Santa Clara, CA 95054

Table of Contents

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is the GlobalProtect client (GlobalProtect App 5.1.5).

The Palo Alto Networks GlobalProtect app provides users with the ability to securely communicate with their internal networks.

The focus on this evaluation is on the TOE functionality supporting the claims in the Protection Profile for Application Software.

The Security Target contains the following additional sections:

1. Product Description
2. Security Problem Definition
3. Security Objectives
4. IT Security Requirements
5. TOE Summary Specification
6. Protection Profile Claims
7. Rationale

1.1 Security Target, TOE and CC Identification

ST Title: Palo Alto Networks GlobalProtect App Version 5.1.5 Security Target
ST Version: 1.0
ST Date: 05/08/2020
TOE Identification: The TOE is available in two versions:

- Windows 10
  - GlobalProtect64-5.1.5.msi
  - SHA-256 checksum: 530C35A1390EEBCFF2F9B8D0781C914561468401D3DE135BADA44D9FB869AE38
- macOS 10.14
  - GlobalProtect-5.1.5.pkg
  - SHA-256 checksum: 102D2EDE71F818FC2F225C6BB1A57D46B806C865A12B9EE333065856E0E2532F

TOE Developer: Palo Alto Networks, Inc.
Evaluation Sponsor: Palo Alto Networks, Inc.

1.2 Conformance Claims

PP Reference: Protection Profile for Application Software Version 1.3 [APPSW]
PP Version: 1.3
PP Date: March 1, 2019

This TOE and ST are conformant to Parts 2 (extended) and 3 (extended) of Common Criteria Version 3.1, Revision 5.

The following NIAP Technical Decisions apply to this PP, and have been accounted for in the ST development:

- 0416 – Correction to FCS_RBG_EXT.1 Test Activity
- 0427 – Reliable Time Source
- 0437 – Supported Configuration Mechanism
- 0434 – Windows Desktop Application Test
- 0444 – IPsec Selections
- 0445 – User Modifiable File Definition
- 0465 – Configuration Storage for .NET Apps
- 0486 – Removal of PP-Module for VPN Clients from allowed with list
- 0495 – FIA_X509_EXT.1.2 Test Clarification
- 0498 – Application Software PP Security Objectives and Requirement Rationale
- 0505 – Clarification of revocation testing under RFC6066
- 0510 – Obtaining random bytes from for iOS/macOS

PP Reference: Functional Package for Transport Layer Security (TLS) [PKGTLS]
PP Version: 1.1
PP Date: February 12, 2019

The TOE and ST is package-name conformant to [PKGTLS].

The following NIAP Technical Decisions apply to this PP, and have been accounted for in the ST development:

- 0442 – Updated TLS Ciphersuites for TLS package
- 0469 – Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1
    - *Note: This Technical Decision is not applicable to the TOE as the TOE does not claim FCS_TLSS_EXT.1.1*
- 0499 – Testing with pinned certificates
- 0513 – CA Certificate loading

1.3 Conventions

The following conventions have been applied to this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a number in parentheses placed at the end of the component. For example, FDP_ACC.1 (1) and FDP_ACC.1 (2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement, (1) and (2).

  - Assignment: allows the specification of an identified parameter. Assignments are indicated using italicized and are surrounded by brackets (e.g., [*assignment*]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold and are surrounded by brackets (e.g., [**selection**]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… some **big** things …"). Note that 'cases' that are not applicable in a given SFR have simply been removed without any explicit identification.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.
- The ST does not highlight operations that have been completed by the PP and EP authors.

1.3.1 Terminology

The following terms and abbreviations are used in this ST:

1.3.2 Acronyms

AES	Advanced Encryption Standard
CBC	Cipher-Block Chaining
CC	Common Criteria for Information Technology Security Evaluation
CEM	Common Evaluation Methodology for Information Technology Security
CM	Configuration Management
CPU	Central Processing Unit
DH	Diffie-Hellman
EEPROM	Electrically Erasable Programmable Read-Only Memory
EP	Extended Package
EST	Enrollment over Secure Transport
FIA	Identification and Authentication CC Class
FIPS	Federal Information Processing Standard
FMT	Security Management CC Class
FSP	Functional Specification
FTP	File Transfer Protocol
GUI	Graphical User Interface
HMAC	Hashed Message Authentication Code
HTTP(S)	Hypertext Transfer Protocol (Secure)
IKE	Internet Key Exchange
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
NDPP	Protection Profile for Network Devices
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
PP	Protection Profile

QoS	Quality of Service
REST	Representational State Transfer
RSA	Rivest, Shamir and Adleman (algorithm for public-key cryptography)
SA	Security Association
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SM	Security Management
SMR	Security Management Roles
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSL	Secure Socket Layer Protocol
ST	Security Target
STFF	Stateful Traffic Filter Firewall (EP)
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functions
TSP	TOE Security Policy
UDP	User Data Protection
URL	Uniform Resource Locator
VLAN	Virtual Local Area Network
VM	Virtual Machine
VPN	Virtual Private Network
VPNGW	VPN Gateway (EP)

2. TOE Description

The TOE is the Palo Alto Networks GlobalProtect App that provides users with the ability to access their company network resources via the Palo Alto Networks GlobalProtect Portals and Gateways that have been deployed. The TOE also provides several management functions that includes, for examples, allowing the endpoint user to select their desired gateway, and to collect troubleshooting logs from the TOE. Additional components that interact with the TOE are noted in the TOE Overview.

2.1 TOE Overview

The GlobalProtect app is a software program that runs on the endpoint (desktop/laptop computer) to protect users by using the same security policies that protect the sensitive resources in corporate networks. The GlobalProtect app secures the traffic using TLS and allows users to connect to corporate networks to access company's resources from anywhere in the world (e.g. when users are remote). The TOE runs on either Windows 10 or macOS (minimum version 10.14).

The TOE is a software program as specified in the APPSW, which uses TLS to protect communication as defined in PKGTLS. The TOE interacts with other GlobalProtect components, which include the Palo Alto Networks GlobalProtect Portal and Gateway.

The Palo Alto Next Generation Firewall provides the GlobalProtect Portal, which provides details for the GlobalProtect infrastructure. Every client system that participates in the GlobalProtect network receives configuration information from the portal, including information about available gateways as well as any client certificates that may be required to connect to the Palo Alto Next Generation Firewall GlobalProtect Gateways. The GlobalProtect gateways provide security enforcement for traffic from GlobalProtect apps.

Once connected, user and host information is sent to the GlobalProtect gateway, which identifies the identity of the operator that is connecting along with details of the host via the host profile (e.g. antivirus definitions installed, security patches, etc.).

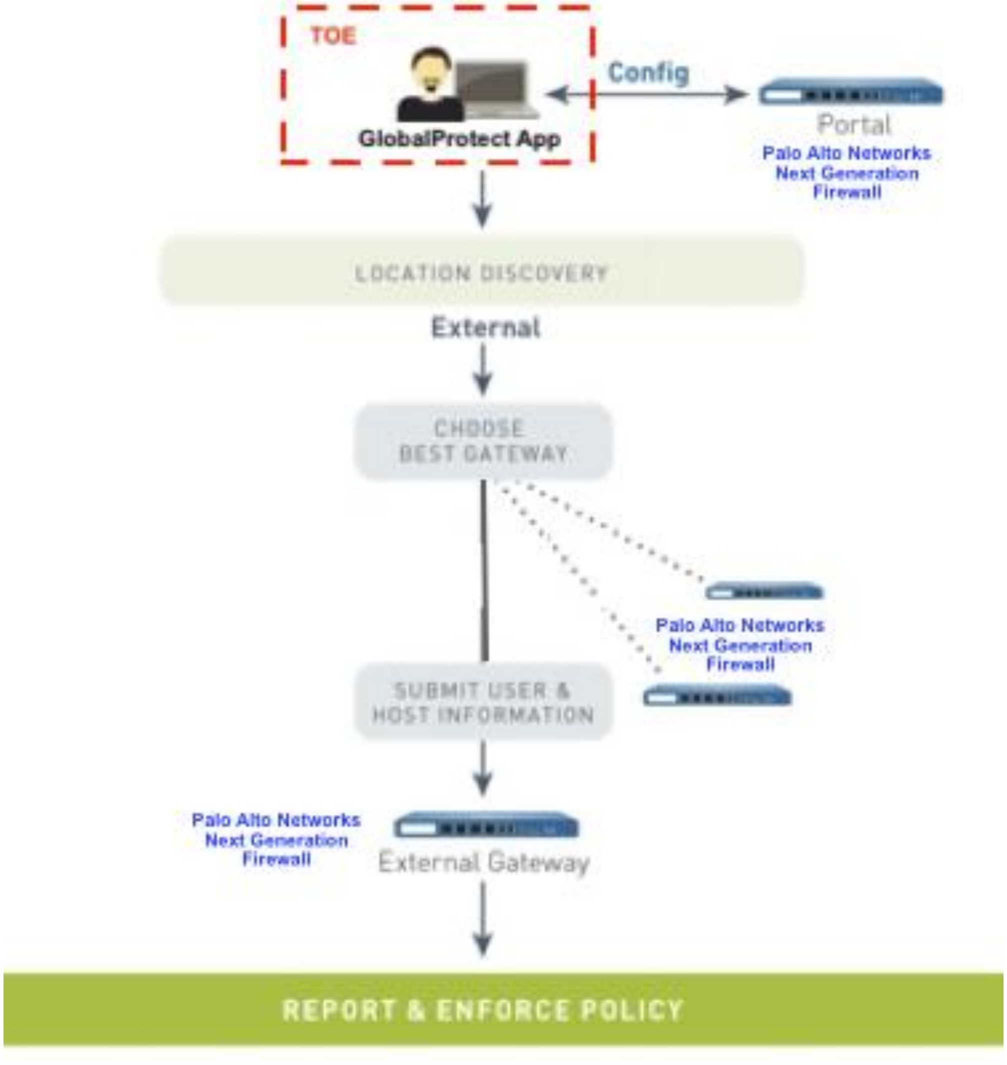
TOE
GlobalProtect App
Config
Portal
Palo Alto Networks Next Generation Firewall
LOCATION DISCOVERY
External
CHOOSE BEST GATEWAY
Palo Alto Networks Next Generation Firewall
SUBMIT USER & HOST INFORMATION
Palo Alto Networks Next Generation Firewall
External Gateway
REPORT & ENFORCE POLICY

Figure 1 – GlobalProtect app

2.2 TOE Architecture

The TOE is a software solution that is comprised of items listed in Section 2.2.1 and 2.2.2. The software is available for download from the Palo Alto Networks support site.

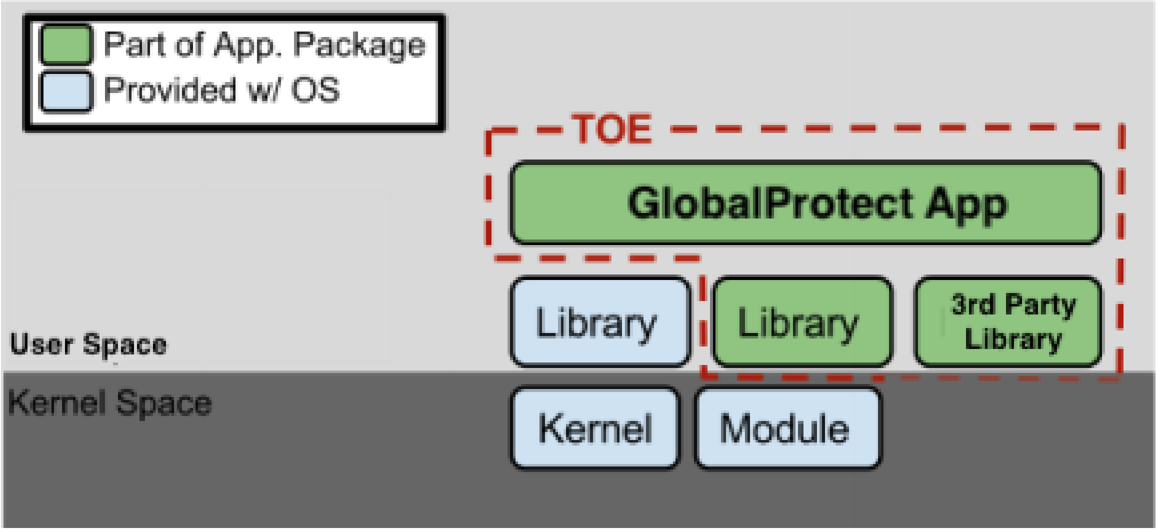

Figure 2 - TOE Architecture

2.2.1 Physical Boundaries

The physical boundary of the TOE is the GlobalProtect app installed and running on a supported platform (i.e. Windows or macOS).

2.2.1.1 Software Requirements

The TOE runs on a desktop operating system that includes macOS version 10.14+ or Windows 10 that communicates with a Palo Alto Networks Next Generation Firewall that utilizes PAN-OS 9.0 or later.

2.2.1.2 Hardware Requirements

The TOE must be installed on either a desktop/laptop computer with macOS or Windows 10. The GlobalProtect Portal and Gateway reside on a Palo Alto Networks Next Generation Firewall. The Palo Alto Networks Next Generation Firewall is covered in a separate evaluation.

Minimum hardware: 256 MB RAM (minimum); 100 MB of disk space (minimum)

The TOE was installed and tested on the following platforms.

- Windows 10 Pro 1909 - Processor: Intel Core i7-4700MQ (Haswell microarchitecture)
- MacMini MacOS version 10.14.6 - Processor: Intel Core i5-8500B (Coffee Lake microarchitecture)

2.2.2 Logical Boundaries

This section summarizes the security function provided by the TOE:

- Cryptographic support
- User data protection
- Identification and Authentication
- Security Management
- Privacy
- Protection of the TSF
- Trusted path/channels

2.2.2.1 Cryptographic support

The TOE implements NIST validated cryptographic algorithms that provide key management, random bit generation, encryption/decryption, digital signature and cryptographic hashing and keyed-hash message authentication features in support of cryptographic protocols such as TLS. In order to utilize these features, the TOE must be configured in FIPS-CC mode.

GlobalProtect App includes algorithms that are covered by CAVP certificates that are noted in this document. In addition, the TOE also relies on the underlying platforms Windows 10 and macOS. **Table 4** contains information regarding all the keys included and utilized by the TOE.

2.2.2.2 User data protection

The TOE restricts its access to only using network connectivity when it is needed to communicate to the Palo Alto Networks Gateway or Portal. Other functionality on the host platform such as its camera, Bluetooth, USB, or microphone are not needed. The TOE does not store any sensitive data in non-volatile memory.

2.2.2.3 Identification and authentication

The TOE authenticates the X.509 certificate of the Palo Alto Networks GlobalProtect Gateway/Portal as part of establishing a TLS connection.

2.2.2.4 Security Management

The TOE provides access to the security management features using an interface on a general-purpose computer. Security management operations are provided to the user of the TOE. A user is able to perform security management by configuring necessary items such as assigning the Palo Alto Networks GlobalProtect Portal and Gateway that the TOE will use for its connections. It also provides the user with the ability to collect troubleshooting logs, configure gateway and portal, check the current version, check for updates, and to enable/disable the transmission of information regarding the system's hardware/software or configuration. The TOE relies on the OS' network ports (i.e. ethernet ports) for communication and management capabilities.

In order to install or uninstall the TOE, the user is required to have platform administrator privileges.

2.2.2.5 Privacy

The TOE does not transmit PII over a network.

2.2.2.6 Protection of the TSF

The TOE implements a variety of functions to ensure that it is protected against corruption. These include utilizing platform APIs, memory mapping, and stack-based buffer overflow protection. Palo Alto Networks provides customers with a means of updating their TOE using trusted updates. These trusted updates are securely delivered and installed using protection mechanisms such as TLS, and by using approved digital signature methods. All of these updates are properly signed using RSA 2048 with SHA-256. The trusted update site also provides a checksum of the updates that can be used for additional verification before it is utilized.

2.2.2.7 Trusted path/channels

The TOE protects communication between itself as the endpoint and other networks using TLS. TLS 1.2 is utilized to encrypt all data that is passed from the TOE to other components (i.e. Palo Alto Networks GlobalProtect Portals and Gateways).

2.3 TOE Documentation

Palo Alto Networks, Inc. has several documents that provide users with information regarding the installation, and the included security features.

For GlobalProtect App 5.1.5, these documents include the following:

- Palo Alto Networks GlobalProtect App Version 5.1.5 Security Target, [This document]
- Palo Alto Networks GlobalProtect App User Guide Version 5.1, June 4, 2020 (Last Updated)
- Palo Alto Networks Common Criteria Evaluation Configuration Guide (CCECG) GlobalProtect App 5.1.5, May 8, 2020

3 Security Problem Definition

This security target includes by reference the Security Problem Definition (composed of organizational policies, threat statements, and assumption) from [APPSW].

In general, the [APPSW] has presented a Security Problem Definition appropriate for software applications, and as such, is applicable to the TOE.

The following threats are directly from the [APPSW]:

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.

The following assumptions are made as drawn directly from the [APPSW]:

A. PLATFORM

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

A. PROPER_USER

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

A. PROPER_ADMIN

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

4 Security Objectives

The sections below identify the security objectives for the TOE and for the operational environment. These security objectives identify the responsibilities of the TOE and the operational environment in meeting security needs.

4.1 Security Objectives for the TOE

The Security Objectives below are defined in the APPSW.

O.INTEGRITY

Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom, if ever, shipped without errors. The ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options.

O.QUALITY

To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs.

O.MANAGEMENT

To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII.

O.PROTECTED_STORAGE

To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data.

O.PROTECTED_COMMS

To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application.

4.2 Security Objectives for the Operational Environment

The Security Objectives below are defined in the APPSW.

OE.PLATFORM

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

OE.PROPER_USER

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

OE.PROPER_ADMIN

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

5 IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the following Protection Profiles (PP):

- *Protection Profile for Application Software, Version 1.3, 1 March 2019* [APPSW],
- *Functional Package for Transport Layer Security (TLS), Version 1.1 [PKGTLS]*

The SARs are the set of SARs specified in [APPSW].

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the [APPSW] and [PKGTLS]. The [APPSW] and [PKGTLS] define all the extended SFRs (*_EXT.1) and since they are not redefined in this ST, the [APPSW] and [PKGSTLS] should be consulted for more information in regard to those CC extensions.

5.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the TOE.

Table 1 TOE Security Functional Components

Requirement Class	Requirement Component
FCS: Cryptographic Support	FCS_CKM_EXT.1: Cryptographic Key Generation Services
	FCS_CKM.1(1): Cryptographic Asymmetric Key Generation
	FCS_CKM.2: Cryptographic Key Establishment
	FCS_RBG_EXT.1 Random Bit Generation Services
	FCS_RBG_EXT.2: Random Bit Generation from Application
	FCS_STO_EXT.1 Storage of Credentials
	FCS_TLS_EXT.1 TLS Protocol
	FCS_TLSC_EXT.1 TLS Client Protocol
	FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication
	FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension
	FCS_COP.1.1(1): Cryptographic Operation – Encryption/Decryption
	FCS_COP.1.1(2): Cryptographic Operation – Hashing
	FCS_COP.1.1(3): Cryptographic Operation -- Signing
	FCS_COP.1.1(4): Cryptographic Operation – Keyed-Hash Message Authentication
FDP: User Data Protection	FDP_DEC_EXT.1 Access to Platform Resources
	FDP_NET_EXT.1 Network Communications
	FDP_DAR_EXT.1 Encryption of Sensitive Application Data
FIA: Identification and Authentication	FIA_X509_EXT.1 X.509 Certificate Validation
	FIA_X509_EXT.2 X.509 Certificate Authentication
FMT: Security Management	FMT_MEC_EXT.1 Supported Configuration Mechanism
	FMT_CFG_EXT.1 Secure by Default Configuration

Requirement Class	Requirement Component
	FMT_SMF.1 Specification of Management Functions
FPR: Privacy	FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information
FPT: Protection of the TSF	FPT_API_EXT.1 Use of Supported Services and APIs
	FPT_AEX_EXT.1 Anti-Exploitation Capabilities
	FPT_TUD_EXT.1 Integrity for Installation and Update
	FPT_TUD_EXT.2 Integrity for Installation and Update
	FPT_LIB_EXT.1 Use of Third Party Libraries
	FPT_IDV_EXT.1 Software Identification and Versions
FTP: Trusted Path/Channels	FTP_DIT_EXT.1 Protection of Data in Transit

5.2.1 Cryptographic Support (FCS)

5.2.1.1 – Cryptographic Key Generation Services (FCS_CKM_EXT.1)

FCS_CKM_EXT.1.1 The application shall [

- **implement asymmetric key generation**

].

5.2.1.2 – Cryptographic Asymmetric Key Generation (FCS_CKM.1(1))

FCS_CKM.1.1(1) The application shall [

- **implement functionality**

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

- **[ECC schemes]** *using* **[“NIST curves” P-256, P-384 and [selection: P-521]** *that meet the following:* **[FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4]**

].

5.2.1.3 – Cryptographic Key Establishment (FCS_CKM.2)

FCS_CKM.2.1 The application shall [**implement functionality**] to perform cryptographic key establishment in accordance with a specified cryptographic key establishment method:
[

- **[Elliptical curve-based key establishment schemes]** *that meets the following*: [**NIST Special Publication 800-56A,**

"Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"]

].

5.2.1.4 – TLS Protocol (FCS_TLS_EXT.1)

FCS_TLS_EXT.1 The product shall implement [

- **TLS as a client**

].

5.2.1.5 – TLS Client Protocol (FCS_TLSC_EXT.1)

FCS_TLSC_EXT.1.1 The product shall implement TLS 1.2 (RFC 5246) and [**no earlier TLS versions**] as a client that supports the cipher suites [

- **TLS_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 [

- **mutual authentication**

].

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

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

5.2.1.6 – TLS Client Support for Mutual Authentication (FCS_TLSC_EXT.2)

FCS_TLSC_EXT.2.1 The product shall support mutual authentication using X.509v3 certificates.

5.2.1.7 – TLS Client Support for Supported Groups Extension (FCS_TLSC_EXT.5)

FCS_TLSC_EXT.5.1 The product shall present the Supported Groups Extension in the Client Hello with the supported groups [

- **secp256r1**
- **secp384r1**
- **secp521r1**

].

5.2.1.8 – Cryptographic Operation – Encryption/Decryption (FCS_COP.1(1))

FCS_COP.1.1(1) The **application** shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm [

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

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

5.2.1.9 – Cryptographic Operation – Hashing (FCS_COP.1(2))

FCS_COP.1.1(2) The **application** shall perform *cryptographic hashing* services in accordance with a specified algorithm [

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

] and message digest sizes [

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

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

5.2.1.10 – Cryptographic Operation – Signing (FCS_COP.1(3))

FCS_COP.1.1(3) The **application** shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

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

] .

5.2.1.11 – Cryptographic Operation – Keyed-Hash Message Authentication (FCS_COP.1(4))

FCS_COP.1.1(4) The **application** shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm

- HMAC-SHA-256

and [

- SHA-1
- **SHA-384**

] with key sizes [*256*, *160*, *384*] and message digest sizes 256 and [**160, 384**] bits that meet the following FIPS Pub 198-1 *The Keyed-Hash Message Authentication Code* and FIPS Pub 180-4 *Secure Hash Standard.*

5.2.1.12 – Random Bit Generation Services (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1 The application shall [

- **implement DRBG functionality**

] for its cryptographic operations.

5.2.1.13 – Random Bit Generation Services (FCS_RBG_EXT.2)

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

FCS_RBG_EXT.2.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from a platform-based DRBG and [

- **no other noise source**

] with a minimum of [

- **256 bits**

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

5.2.1.14 – Storage of Credentials (FCS_STO_EXT.1)

FCS_STO_EXT.1.1 The application shall [

- **invoke the functionality provided by the platform to securely store [*CA certificate*s, *user certificates, RSA private keys, ECDSA private keys]***

] to non-volatile memory.

5.2.2 User Data Protection (FDP)

5.2.2.1 – Access to Platform Resources (FDP_DEC_EXT.1)

FDP_DEC_EXT.1.1 The application shall restrict its access to [

- **network connectivity**

].

FDP_DEC_EXT.1.2 The application shall restrict its access to [

- **no sensitive information repositories**

].

5.2.2.2 – Network Communications (FDP_NET_EXT.1)

FDP_NET_EXT.1.1 The application shall restrict network communication to [

- **user-initiated communication for *[connections to Palo Alto Networks Next Generation Firewall Gateways and Portals]***.

].

5.2.2.3 – Encryption of Sensitive Application Data (FDP_DAR_EXT.1)

FDP_DAR_EXT.1.1 The application shall [

- **not store any sensitive data**

] in non-volatile memory.

5.2.3 Security Management (FMT)

5.2.3.1 – Supported Configuration Mechanism (FMT_MEC_EXT.1)

FMT_MEC_EXT.1.1 The application shall [

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

5.2.3.2 – Secure by Default Configuration (FMT_CFG_EXT.1)

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.

5.2.3.3 – Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions [

- **enable/disable the transmission of any information describing the system's hardware, software, or configuration**
- ***[setting gateway and portal addresses***
- ***collecting troubleshooting logs***
- ***check for updates***
- ***querying the current version of the TOE]***

]

5.2.4 Privacy

5.2.4.1 – User Consent for Transmission of Personally Identifiable Information (FPR_ANO_EXT.1)

FPR_ANO_EXT.1.1 The application shall [

- **not transmit PII over a network**

].

5.2.5 Protection of the TSF (FPT)

5.2.5.1 – Use of Supported Services and APIs (FPT_API_EXT.1)

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

5.2.5.2 – Anti-Exploitation Capabilities (FPT_AEX_EXT.1)

FPT_AEX_EXT.1.1 The application shall not request to map memory at an explicit address expect for [*no exceptions*].

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

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

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

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

5.2.5.3 – Integrity for Installation and Update (FPT_TUD_EXT.1)

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

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

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

FPT_TUD_EXT.1.4 The application installation package and its updates shall be digitally signed such that its platform can cryptographically verify them prior to installation.

FPT_TUD_EXT.1.5 The application is distributed [**as an additional software package to the platform O*S***].

5.2.5.4 – Integrity for Installation and Update (FPT_TUD_EXT.2)

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

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

5.2.5.5 – Use of Third Party Libraries (FPT_LIB_EXT.1)

FPT_LIB_EXT.1.1 The application shall be packaged with only [*OpenSSL, OESIS*]

5.2.5.6 – Software Identification and Versions (FPT_IDV_EXT.1)

FPT_IDV_EXT.1.1 The application shall be versioned with [***[GlobalProtect software version]***].

5.2.6 Trusted Path/Channel (FTP)

5.2.6.1 – Protection of Data in Transit (FTP_DIT_EXT.1)

FTP_DIT_EXT.1.1 The application shall [

- **encrypt all transmitted [data] with [TLS as defined in the TLS package]**

] between itself and another trusted IT product.

5.2.7 Identification and Authentication (FIA)

5.2.7.1 – X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1 The application shall [**implement functionality**] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path verification.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The application shall validate the revocation status of the certificate using [**the Online Certificate Status Protocol (OCSP)**

as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3].

- The application shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
    - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the 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.

5.2.7.2 – X.509 Certificate Authentication (FIA_X509_EXT.2)

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

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

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are included by reference to [APPSW].

Table 2 Assurance Components

Requirement Class	Requirement Component
ASE: Security Target	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition
	ASE_INT.1 ST introduction
	ASE_OBJ.1 Security objectives
	ASE_REQ.1 Security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ADV: Development	ADV_FSP.1 Basic functional specification
AGD: Guidance Documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-Cycle Support	ALC_CMC.1 Labelling of the TOE
	ALC_CMS.1 TOE CM coverage
	ALC_TSU_EXT.1 Timely Security Updates
ATE: Tests	ATE_IND.1 Independent testing - conformance
AVA: Vulnerability Assessment	AVA_VAN.1 Vulnerability survey

6 TOE Summary Specification

This chapter describes the security functions:

- Cryptographic support
- User data protection
- Certificate validation
- Security management
- Privacy
- Protection of the TSF
- Trusted path/channels

FCS_CKM.1.1(1), FCS_CKM_EXT.1 (APPSW)

The GlobalProtect app itself does not generate certificates. Platform Administrators are able to set/load client or server certificates into the certificate store of the platform (i.e., keypair generated on the platform) that is running on. During a TLS handshake, the TOE utilizes ECDHE for the key establishment with NIST curves that include P-256, P-384, and P-521 that adhere to the NIST Special Publication 800-56A. For details regarding the algorithms supported and their CAVP certificates, see **Table 4**.

FCS_CKM.2.1 (APPSW)

The TOE implements key establishment methods using elliptical curve key establishment scheme (ECDHE). The curves utilized by the TOE include P-256, P-384, and P-521 as defined in NIST SP 800-56A.

FCS_COP.1(1) (APPSW)

The TOE is able to encrypt/decrypt using AES-CBC mode (as defined in NIST SP 800-38A) and AES-GCM mode (as defined in NIST SP 800-38D) with key sizes 128-bits and 256-bits. Corresponding CAVP certificates for these algorithms are present in **Table 4**.

FCS_COP.1(2) (APPSW)

The TOE uses hash functions that include SHA-1, SHA-256 and SHA-384 as defined in FIPS 180-4. The digest sizes include 160-bits, 256-bits, and 384-bits that are compliant with FIPS 180-4. The hashing capabilities are utilized for digital signature verification and generation and data integrity checks. SHA-1 is not used for generating digital signatures as noted in SP 800-131A but is only used for verification for legacy purpose. The TOE uses SHA-256 and SHA-384 hashing as part of generating digital signatures. SHA-1 is used as part of the software integrity power-up test. Corresponding CAVP certificates for these algorithms are present in **Table 4**.

FCS_COP.1(3) (APPSW)

Both RSA and ECDSA schemes are used for TLS functions with approved key sizes. These include RSA 2048-bits, 3072-bits, and 4096-bits. For ECDSA, they include the curves P-256, P-384, and P-521. During TLS handshakes, these certificates are used for peer authentication to verify the server's identity. These certificates are also used by the TOE to present its identity as a client when connecting to a Palo Alto Networks Gateway. Corresponding CAVP certificates and the relevant schemes for these algorithms are present in **Table 4**.

FCS_COP.1(4) (APPSW)

The TOE supports the use of a Keyed-Hash Message Authentication algorithms that include HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-384. They include key sizes of 160 bits, 256-bits, and 384-bits respectively. The HMAC-SHA functions are used as part of the TOE's integrity check (HMAC-SHA-1) to ensure that it has not been tampered, and is additionally used as part of the TLS handshake (HMAC-SHA-256 and HMAC-SHA-384). Corresponding CAVP certificates for these algorithms are present in **Table 4**.

FCS_TLSC_EXT.1, FCS_TLSC_EXT.2 (PKGTLS), FTP_DIT_EXT.1 (APPSW)

All data that is transmitted between the GlobalProtect app and the Palo Alto Networks Gateway and Portal are encrypted using TLS. When the TOE is establishing a TLS session, it checks the reference identifier that has been specified by the user via the GlobalProtect app. These reference identifiers include IP addresses, and are checked when looking at the Common Name or in the Subject Alternative Name. The TOE supports the handling of wildcards if a certificate is presented with one in it. Certificate pinning is not supported.

The TOE shall not establish a trusted channel if the server certificate is invalid – no exceptions. During the TLS handshake with connections to the Palo Alto Networks Gateway and Portal (both acting as the server), the TOE presents the following cipher suites in its Client Hello. The TOE is only a client, and does not act as a server in any connection. TLS 1.2 is the only version of TLS supported by the TOE.

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

During the negotiation of the TLS handshake, X.509v3 certificates are used to verify the server's identity. Additionally, client certificates can be set on the GlobalProtect app to support mutual authentication. For the cipher suites noted above, the client hello extension supports secp256r1, secp384r1, and secp521r1 curves.

FCS_TLSC_EXT.5 (PKGTLS)

The TOE presents the Supported Groups Extension in its Client Hello that includes the following groups: secp256r1, secp384r1, and secp521r1. No other groups are supported.

FCS_RBG_EXT.1, FCS_RBG_EXT.2 (APPSW)

The TOE implements DRBG functionality using the CTR_DRBG in AES mode by default. The DRBG is seeded using the Windows 10 or macOS DRBG, which provides a minimum of 256 bits of entropy. A description of the noise sources for the operating systems are noted below.

The entropy pool for Windows 10 is populated using the following values:

- An initial entropy value provided by the Windows OS Loader at boot time.
- The values of the high-resolution CPU cycle counter at times when hardware interrupts are received.
- Random values gathered from the Trusted Platform Module (TPM), if one is available on the system.
- Random values gathered by calling the RDRAND CPU instruction, if supported by the CPU.

For macOS, the deterministic random bit generators are seeded by /dev/random. The /dev/random generator is a true random number generator that obtains entropy from interrupts generated by the devices and sensors attached to the system and maintains an entropy pool. The NDRNG feeds entropy from the pool into the DRBG on demand.

FCS_STO_EXT.1 (APPSW)

The TOE uses the functionality provided by the platform in order to securely store X.509 certificates that are used for connections to the Palo Alto Networks GlobalProtect Gateway/Portal. The platform provides the necessary security in order to protect these items.

For macOS, the necessary certificates are stored within the Keychain while on Windows, certificates are stored within the Windows Certificate Store.

The TOE's keys/credentials are noted in **Table 3**.

Table 3 - Keys and Credentials

Key	Description/Usage	Storage
CA Certificates	Used to extend trust for certificates (ECDSA – P-256/384/521) (RSA – 2048/3072/4096 bits)	OS' key store
RSA Public Keys	RSA public keys managed as certificates for the verification of signatures, establishment of TLS, and peer authentication. (RSA 2048/3072/4096 bits)	OS' key store
RSA Private Keys	RSA Private key used for authentication, and signature generation (RSA 2048, 3072, or 4096 bits)	OS' key store
ECDSA Public Keys	ECDSA public keys managed as certificates for the verification of signatures, establishment of TLS, and peer authentication. (P-256/384/521)	OS' key store
ECDSA Private Keys	ECDSA Private key used for authentication, and signature generation (P-256, P-384 or P-521)	OS' key store

FDP_DAR_EXT.1.1, FDP_DEC_EXT.1, FDP_NET_EXT.1 (APPSW)

The GlobalProtect app does not store any sensitive data in non-volatile memory. During the configuration of the TOE, the user is not able to enter any sensitive data. When an user is initiating connections to the Palo Alto Networks Gateway or Portal, they are required to enter their authentication data for each new session that includes the username and password that is defined from the Palo Alto Networks Next Generation Firewall Gateway/Portal for the connection to succeed. These credentials are not stored or managed by GP.

The application restricts its access to only using network connectivity when it is needed to communicate to the Palo Alto Networks Gateway or Portal. Other functionality on the host platform such as its camera, Bluetooth, USB, or microphone are not needed.

FMT_MEC_EXT.1.1, FMT_CFG_EXT.1, FMT_SMF.1 (APPSW)

When the TOE is configured, it is required that the platform administrator follow the rules defined in the administrator guide to properly set the correct configuration. If they are not followed, the GlobalProtect app will be active in non-FIPS-CC mode. The configuration of the TOE must be completed by a platform administrator that is present at the endpoint on which the TOE resides as it will need administrator privileges to perform the installation of the software. There are no default credentials that are used or included with the TOE during its configuration.

If the TOE is installed on a Windows 10 environment, it is required that the platform OS' FIPS mode be enabled. FIPS mode is also required for the macOS platform, but this is enabled by default for macOS releases 10.12 and newer.

As noted in the Common Criteria Evaluated Configuration Guide, a platform administrator setting the TOE on a Windows 10 environment is required to launch the Windows Registry and make the proper edits there to set FIPS-CC mode.

For the macOS configuration, a platform administrator is required to edit the relevant plist file to set the FIPS-CC mode of the TOE. This file is located in the platform's Library folder. Detailed instructions on how to set the required settings for enabling FIPS-CC mode for the GlobalProtect app on macOS is included in the CCECG.

Once the TOE has been properly initialized into FIPS-CC mode, the TOE will have the ability to connect to the Palo Alto Networks Gateways provided by the Palo Alto Networks Next Generation Firewalls. The TOE provides several management functions that includes the following that can be performed by the user:

- Enable/disable the transmission of any information describing the system's hardware, software, or configuration
- Setting gateway and portal addresses
- Check for updates
- Collecting troubleshooting logs (i.e. GlobalProtect app system logs for the application, self-test results, connection details)
- Querying the current version of the TOE

By default, the TOE includes file permissions that protect the TOE's binary and data files from modification from normal unprivileged users. The TOE also includes an integrity check for itself to ensure that no malicious activity occurs.

FPR_ANO_EXT.1 (APPSW)

The GlobalProtect app does not transmit personally identifiable information about an individual. While the TOE may use client certificates to identify itself to the Palo Alto Networks GlobalProtect Gateway, it does not include sensitive information such as financial records, medical history, or social security numbers that could be used to identify an individual.

FPT_API_EXT.1 (APPSW)

The TOE includes the use of platform APIs for Windows and macOS. These are noted in Appendix A.

FPT_AEX_EXT.1 (APPSW)

The TOE automatically enables ASLR when the application is compiled on Windows 10 (/DYNAMICBASE link flag) or macOS (-pie link flag), and stack-based buffer overflow protection is enabled by default (compiled with /GS flag). There is no administrator intervention required to set this item. The GlobalProtect app does not request any memory mapping at an explicit address. The TOE does not allocate any memory region with both write and execute permissions; users shall also not write user-modifiable files to directories that contain executable files unless they are explicitly told to do so.

The GlobalProtect app is designed to be compatible with the security features that are provided by the platform (Windows and macOS) vendor that is it installed on.

FPT_TUD_EXT.1 (APPSW)

The TOE has specific versions, which can be queried by the user via the TOE's interface. New versions of the TOE are created by Palo Alto Networks, which an administrator can retrieve to update the current version of the TOE. During the installation process, a digital signature verification check is automatically performed to verify that the update has not been modified. All new versions of the GlobalProtect app are digitally signed by Palo Alto Networks using RSA 2048 with SHA-256.

Updates are available at https://support.paloaltonetworks.com or can be retrieved from the GlobalProtect Portal if a new version has been downloaded and activated on the Palo Alto Networks Next Generation Firewall. The TOE cannot update its own binary code – it relies on the administrator to download and install the new version available.

FPT_TUD_EXT.2 (APPSW)

The following package formats are used for the GlobalProtect installation file:

- Windows 10: *GlobalProtect64-5.1.5.msi*
- macOS: *GlobalProtect-5.1.5.pkg*

The TOE is packaged such that the uninstall of the software results in complete zeroization of the TOE automatically. All files are removed from the platform when this uninstall process is initiated. Before files are uninstalled, they are overwritten with a random pattern, and then zeroized.

For Windows, this is done by selecting the program via the Control Panel and selecting uninstall. On macOS, zeroization is performed by selecting the GlobalProtect app in the Applications section of macOS' Finder and moving it to trash. The TOE overwrites files with random bytes first before it is removed from the system.

FPT_LIB_EXT.1 (APPSW)

The TOE utilizes OpenSSL for its crypto functions and OESIS to provide endpoint security detection service in both macOS and Windows platforms. This library is checked for its integrity during the installation/initialization period to ensure that it has not been tampered with, and that the necessary procedures are followed to place this library in its required FIPS mode.

FPT_IDV_EXT.1 (APPSW)

Palo Alto Networks provides a version control system for its software components. The TOE has a unique software versioning that identifies major versions and their subsequent maintenance releases in the following form: <major>.<minor>.<maintenance release>. Major and minor releases introduce new major and minor features for the product, and additional maintenance releases (e.g. 5.1.0, 5.1.1[1]) are released on a regular cadence to fix issues identified with the major release.

FIA_X509_EXT.1, FIA_X509_EXT.2 (APPSW)

The GlobalProtect app implements the ability to perform certificate path validation on the certificate chain that is presented to it by the Palo Alto Networks GlobalProtect Gateway or Portal. The certificate path validation begins with the identity certificate presented by the Gateway or Portal, and then proceeds in checking the intermediate CA certificate(s) until it reaches the trusted root certificate issued in the platform OS trust store. Only root certificates stored here are used and trusted by the TOE. On Windows platform, use the Certificates Snap-In (from the MMC) and on MacOS platform, use the Keychain to install the certificate. The following steps are performed for each certificate in the path:

- The public key algorithm/parameters are checked (i.e. RSA/ECDSA key sizes meet FIPS-CC requirements)
- The certificate is checked to make sure it is not expired (i.e. validity period of the certificate must be proper)
- The certificate is checked to make sure it is not revoked using either CRL/OCSP
- The issuer name is checked to ensure that it matches the subject name of the previous certificate in the chain

[1] There is also an internal build number which may be displayed. This is used by the vendor for internal tracking only.

- The certificate is checked that it terminates with a trusted CA certificate and that all CA certificate have the basicConstraints extension present (and set to TRUE)
- The extendedKeyUsage field is checked such that OCSP certificates and server certificates contain the correct OID (e.g. OCSP Signing purpose and Server Authentication purpose)
- The key usage extension of the certificate is checked to make sure that it is allowed to sign certificates
- Path lengths are checked to ensure it does not exceed any maximum path length inserted

Certificates that are presented to the TOE must meet the x509v3 requirements as defined in RFC 5280 for TLS. If there are any issues with the certificate presented (as noted above), the application will not accept the certificate and reject the connection. A log message will be generated, and an administrator will be required to address the problem noted in order for the connection to succeed. The TOE will also display an error window with the failure reason and the option to continue is greyed out (i.e. unable to be selected). In FIPS-CC mode, the option to continue or override based on the administrator discretion is disabled.

The TOE also supports the revocation checking of the certificate presented using either OCSP or CRL (as specified in RFC 2560 and RFC 5280 Section 6.3). In the event that the certificate is revoked following a check of its status, the TOE will reject the connection, and not allow the connection to continue. In the event that OCSP/CRL can’t be reached, the administrator is provided with a warning message that the revocation status cannot be checked or determined along with the option to proceed with the connection as permitted by FIA_X509_EXT.2.2.

Certificates are not used for email encryption, or server certificates presented for EST.

ALC_TSU_EXT.1 (APPSW)

The TOE is regularly updated with maintenance releases once a major release is made available to the public. These maintenance releases include various bug fixes to improve product features and to address any security vulnerabilities that may have come up in previous versions. When a new version is available, users are notified via an email from Palo Alto Networks with the specific version published. These versions are also displayed on Palo Alto Networks’ Customer Support page (https://support.paloaltonetworks.com). An updated version of the product is made available approximately every 42 – 60 days.

The support portal provides users the ability to download new versions of the software. This portal also includes links to the Palo Alto Networks Release Notes that highlight all the changes included in the published release. These release notes detail all the bug fixes and security advisories/vulnerabilities that have been addressed. When a user downloads the new version from the support portal there is an option to display the SHA-256 checksum of the file that can be verified again once the file is downloaded. Each file for macOS or Windows is marked with the relevant version of the TOE in the following format:

- Windows 10: *GlobalProtect64-<version>.msi*
- macOS: *GlobalProtect-<version>.pkg*

Palo Alto Networks provides customers with a Security Advisory page for any security vulnerabilities that have been identified in Palo Alto Networks products (https://securityadvisories.paloaltonetworks.com/).

Each vulnerability is given a criticality rating and an updated status on any updates or mitigations regarding each discovered vulnerability. Each vulnerability listing also provides a list of the versions of the product that the vulnerability is known to affect. In the event that a vulnerability has been discovered, Palo Alto Networks provides users with the ability to report them via the Product Security Incident Response Team (PSIRT) via a trusted channel for a website:
(https://securityadvisories.paloaltonetworks.com/Report)

6.1 Cryptographic Algorithms

The following table includes the CAVP certificates obtained for the two operational environments (Windows 10 and macOS).

Equivalency Argument: GlobalProtect version 5.0 and version 5.1 contain and utilize the exact same underlying cryptographic module (OpenSSL), and it is this module that was CAVP validated. Both versions call (via APIs) the same Approved FIPS algorithms with the same Approved key sizes in FIPS-CC mode.

Table 4 - Cryptographic Functions

Function(s)	Standards	Certificates
Asymmetric key generation (FCS_CKM.1(1))		
RSA (2048 bits or greater)	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3	#C1544
ECDSA (P-256, P-384, P-521 curves)	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4	#C1544

Cryptographic key establishment (FCS_CKM.2)		
Elliptic curve-based scheme	NIST Special Publication 800-56A Revision 2, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”	KAS #C1544
Symmetric encryption/decryption (FCS_COP.1(1))		
AES CBC, GCM (128, 256 bits)	FIPS PUB 197 CBC as defined in NIST SP 800-38A GCM as defined in NIST SP 800-38D	AES #C1544
Cryptographic hashing (FCS_COP.1(2))		
SHA-1, SHA-256, SHA-384	FIPS PUB 180-4	SHS #C1544
Cryptographic signature services (FCS_COP.1(3))		
RSA with 2048-bit modulus or greater	FIPS PUB 186-4	RSA #C1544
ECDSA with NIST Curves P-256, P-384, and P-521	FIPS PUB 186-4	ECDSA #C1544
Keyed-hash message authentication (FCS_COP.1(4))		
HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-384	FIPS Pub 198-1 FIPS Pub 180-4	HMAC #C1544 SHS #C1544
Deterministic random bit generation (FCS_RBG_EXT.2)		
CTR_DRBG (AES)	NIST SP 800-90A	DRBG #C1544

7 Protection Profile Claims

This ST is conformant to the [APPSW]. The table below identifies all the security functional requirements within this Security Target.

Table 5 - SFR and Source

Requirement Class	Requirement Component	Source
FCS: Cryptographic Support	FCS_CKM_EXT.1: Cryptographic Key Generation Services	Application Software Protection Profile Functional Package for Transport Layer Security (TLS)(**)
	FCS_CKM.2 Cryptographic Key Establishment	
	FCS_CKM.1(1): Cryptographic Asymmetric Key Generation	
	FCS_RBG_EXT.1 Random Bit Generation Services	
	FCS_RBG_EXT.2: Random Bit Generation Services	
	FCS_STO_EXT.1 Storage of Credentials	
	FCS_TLS_EXT.1 TLS Protocol**	
	FCS_TLSC_EXT.1 TLS Client Protocol**	
	FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication**	
	FCS_TLSC_EXT.5: TLS Client Support for Supported Groups Extension**	
	FCS_COP.1.1(1): Cryptographic Operation – Encryption/Decryption	
	FCS_COP.1.1(2): Cryptographic Operation – Hashing	
	FCS_COP.1.1(3): Cryptographic Operation -- Signing	
	FCS_COP.1.1(4): Cryptographic Operation – Keyed-Hash Message Authentication	
FDP: User Data Protection	FDP_DAR_EXT.1 Encryption of Sensitive Application Data	
	FDP_DEC_EXT.1 Access to Platform Resources	
	FDP_NET_EXT.1 Network Communications	
	FIA_X509_EXT.1 X.509 Certificate Validation	

Requirement Class	Requirement Component	Source
FIA: Identification and Authentication	FIA_X509_EXT.2 X.509 Certificate Authentication	
FMT: Security Management	FMT_CFG_EXT.1 Secure by Default Configuration	
	FMT_MEC_EXT.1 Supported Configuration Mechanism	
	FMT_SMF.1 Specification of Management Functions	
FPR: Privacy	FPR_ANO_EXT.1 User Consent for Transmission of Personally Identifiable Information	
FPR: Protection of the TSF	FPT_AEX_EXT.1 Anti-Exploitation Capabilities	
	FPT_API_EXT.1 Use of Supported Services and APIs	
	FPT_LIB_EXT.1 Use of Third Party Libraries	
	FPT_TUD_EXT.1 Integrity for Installation and Update	
	FPT_TUD_EXT.2 Integrity for Installation and Update	
	FPT_IDV_EXT.1 Software Identification and Versions	
FTP: Trusted Path/Channels	FTP_DIT_EXT.1 Protection of Data in Transit	

• Rationale

This security target includes by reference the [APPSW] Security Problem Definition, Security Objectives, and Security Assurance Requirements. The security target makes no additions to the [APPSW] assumptions. Security functional requirements have been reproduced verbatim with the protection profile operations completed. Operations on the security requirements follow [APPSW] application notes and assurance activities. The security target did not add or remove any security requirements. Consequently, [APPSW] rationale applies and is complete.

Appendix A

Windows 10 APIs used by GlobalProtect

ActivateActCtx

AdjustTokenPrivileges

AppendMenuW

BitBlt

BringWindowToTop

CLSIDFromString

CancelIPChangeNotify

CancelMibChangeNotify2

CertAddEncodedCertificateToStore

CertAddSerializedElementToStore

CertCloseStore

CertDeleteCertificateFromStore

CertDuplicateCertificateContext

CertEnumCertificatesInStore

CertFindCertificateInStore

CertFindChainInStore

CertFreeCertificateChain

CertFreeCertificateContext

CertGetCertificateChain

CertGetCertificateContextProperty

CertGetEnhancedKeyUsage

CertGetIntendedKeyUsage

CertGetNameStringA

CertGetNameStringW

CertNameToStrA

CertNameToStrW

CertOpenStore

CertOpenSystemStoreA

CertOpenSystemStoreW

CertSerializeCertificateStoreElement

ChangeServiceConfig2A

CheckMenuItem

CloseHandle

CloseServiceHandle

GetSystemDirectoryA

GetSystemDirectoryW

GetSystemInfo

GetSystemMenu

GetSystemMetrics

GetSystemTimeAsFileTime

GetTempPathA

GetTextExtentPoint32W

GetTextMetricsW

GetThemeSysSize

GetTickCount

GetTickCount64

GetUserDefaultLCID

GetUserDefaultUILanguage

GetUserNameW

GetUserObjectInformationW

GetUserProfileDirectoryA

GetUserProfileDirectoryW

GetVersion

GetVersionExA

GetVersionExW

GetWindowLongW

GetWindowRect

GetWindowTextW

GetWindowThreadProcessId

GetWindowsDirectoryA

GlobalAlloc

GlobalFree

GlobalMemoryStatus

HeapAlloc

HeapDestroy

HeapFree

HeapReAlloc

HeapSize

WideCharToMultiByte

WinHttpCloseHandle

WinHttpConnect

WinHttpGetIEProxyConfig-ForCurrentUser

WinHttpGetProxyForUrl

WinHttpOpen

WinHttpOpenRequest

WinHttpQueryAuthSchemes

WinHttpQueryDataAvailable

WinHttpQueryHeaders

WinHttpQueryOption

WinHttpReadData

WinHttpReceiveResponse

WinHttpSendRequest

WinHttpSetCredentials

WinHttpSetOption

WinHttpSetStatusCallback

WinHttpSetTimeouts

WinVerifyTrust

WlanCloseHandle

WlanEnumInterfaces

WlanFreeMemory

WlanGetAvailableNetworkList

WlanOpenHandle

WriteFile

_CxxThrowException

_XcptFilter

__C_specific_handler

__CxxFrameHandler3

__RTDynamicCast

__crtCaptureCurrentContext

__crtCapturePreviousContext

__crtGetShowWindowMode

__crtSetUnhandledExceptionFilter

CoCreateInstance

CoInitialize

CoInitializeEx

CoInitializeSecurity

CoSetProxyBlanket

CoTaskMemAlloc

CoUninitialize

CombineRgn

ControlService

ConvertSidToStringSidA

CopyRect

CreateActCtxW

CreateCompatibleBitmap

CreateCompatibleDC

CreateDCW

CreateDirectoryW

CreateEnvironmentBlock

CreateEventA

CreateEventW

CreateFileA

CreateFileMappingA

CreateFileMappingW

CreateFileW

CreateFontIndirectW

CreateFontW

CreateIpForwardEntry

CreateIpForwardEntry2

CreateMenu

CreateMutexW

CreatePersistentTcpPortReservation

CreatePersistentUdpPortReservation

CreatePipe

CreatePolygonRgn

CreatePopupMenu

CreateProcessA

CreateProcessAsUserA

CreateProcessW

CreateRectRgn

ImpersonateLoggedOnUser

InetNtopW

InetPtonW

InitializeCriticalSection

InitializeCriticalSectionAndSpinCount

InitializeCriticalSectionEx

InitializeUnicastIpAddressEntry

InstallHinfSectionW

InternetSetOptionA

InvalidateRect

IsDebuggerPresent

IsIconic

IsProcessorFeaturePresent

IsRectEmpty

IsWindow

IsWindowEnabled

K32GetProcessImageFileNameA

KillTimer

LeaveCriticalSection

LoadBitmapW

LoadCursorW

LoadIconW

LoadImageW

LoadLibraryA

LoadLibraryW

LoadMenuW

LoadResource

LoadUserProfileW

LocalAlloc

LocalFree

LockResource

LookupAccountNameW

LookupPrivilegeValueA

LsaEnumerateLogonSessions

LsaFreeReturnBuffer

LsaGetLogonSessionData

LsaNtStatusToWinError

MapViewOfFile

__crtTerminateProcess

__crtUnhandledException

__crt_debugger_hook

__dllonexit

__getmainargs

__initenv

__iob_func

__set_app_type

__setusermatherr

__wgetmainargs

_access

_amsg_exit

_atoi64

_beginthread

_beginthreadex

_calloc_crt

_cexit

_close

_commode

_configthreadlocale

_endthreadex

_errno

_exit

_fileno

_fmode

_fsopen

_fstat64i32

_ftime64

_getch

_getpid

_gmtime64

_initterm

_initterm_e

_ismbcspace

_local_unwind

_localtime64

_localtime64_s

_lock

CreateRoundRectRgn
CreateServiceA
CreateSolidBrush
CreateTimerQueueTimer
CreateToolhelp32Snapshot
CreateUnicastIpAddressEntry
CredDeleteW
CredFree
CredReadW
CredWriteW
CryptAcquireCertificatePrivateKey
CryptDecodeObject
CryptMsgClose
CryptMsgGetParam
CryptQueryObject
CryptUIDlgViewContext
CryptUIWizImport
DeactivateActCtx
DebugBreak
DecodePointer
DeleteCriticalSection
DeleteDC
DeleteFileA
DeleteFileW
DeleteIpForwardEntry
DeleteIpForwardEntry2
DeleteMenu
DeleteObject
DeletePersistentTcpPortReservation
DeletePersistentUdpPortReservation
DeleteService
DeregisterEventSource
DestroyEnvironmentBlock
DestroyIcon
DhcpRequestParams
DnsFree
DnsQuery_A
DrawAnimatedRects

MessageBoxW
ModifyMenuW
MulDiv
MultiByteToWideChar
NetApiBufferFree
NetUserGetInfo
NotifyAddrChange
NotifyRouteChange
NotifyUnicastIpAddressChange
OffsetRect
OpenEventA
OpenFileMappingA
OpenFileMappingW
OpenMutexW
OpenProcess
OpenProcessToken
OpenSCManagerA
OpenServiceA
OpenServiceW
OutputDebugStringA
OutputDebugStringW
PathAppendW
PathFindFileNameA
PostMessageA
PostMessageW
PostThreadMessageA
Process32First
Process32Next
ProcessIdToSessionId
QueryActCtxW
QueryDosDeviceW
QueryPerformanceCounter
QueryServiceStatusEx
RaiseException
ReadFile
RedrawWindow
RegCloseKey
RegCreateKeyExA

_lseek
_mbscmp
_mbsicmp
_mbsinc
_mbslwr_s
_mbsrchr
_mbsstr
_mktime64
_onexit
_purecall
_read
_recalloc
_setmode
_snprintf
_snprintf_s
_stat64i32
_strdup
_stricmp
_strnicmp
_strupr
_swprintf
_time64
_unlink
_unlock
_vscprintf
_vsnprintf
_vsnprintf_s
_vsnwprintf
_vswprintf_c_l
_wcmdln
_wcsdup
_wcsicmp
_wcsnicmp
_wfopen
_wfopen_s
_wopen
_wrename
_wstat64i32

DrawIcon
DuplicateHandle
DuplicateTokenEx
DwmExtendFrameIntoClientArea
EnableMenuItem
EnableWindow
EncodePointer
EnterCriticalSection
EnumChildWindows
EnumDependentServicesA
EnumProcesses
EnumWindows
ExitProcess
ExpandEnvironmentStringsA
ExpandEnvironmentStringsForUserA
FileTimeToLocalFileTime
FileTimeToSystemTime
FillRect
FindActCtxSectionStringW
FindClose
FindFirstFileA
FindFirstFileW
FindNextFileW
FindResourceW
FindWindowW
FlushConsoleInputBuffer
FormatMessageA
FormatMessageW
FreeLibrary
FreeMibTable
FwpmCalloutDeleteByKey0
FwpmEngineClose0
FwpmEngineOpen0
FwpmFilterAdd0
FwpmFilterCreateEnumHandle0
FwpmFilterDeleteByKey0
FwpmFilterDestroyEnumHandle0
FwpmFilterEnum0

RegCreateKeyExW
RegCreateKeyW
RegDeleteKeyA
RegDeleteKeyW
RegDeleteValueA
RegDeleteValueW
RegEnumKeyA
RegEnumKeyExA
RegEnumKeyExW
RegFlushKey
RegNotifyChangeKeyValue
RegOpenKeyA
RegOpenKeyExA
RegOpenKeyExW
RegQueryValueExA
RegQueryValueExW
RegSetKeyValueA
RegSetKeyValueW
RegSetValueExA
RegSetValueExW
RegisterEventSourceW
RegisterServiceCtrlHandlerExA
RegisterWindowMessageW
ReleaseDC
ReleaseMutex
ReportEventW
ResetEvent
RevertToSelf
RtlVirtualUnwind
SHAppBarMessage
SHDeleteValueA
SHGetFolderPathW
SHGetValueA
SHGetValueW
SHSetValueA
SearchPathA
SelectObject
SendMessageTimeoutA

_wtoi
_wunlink
abort
asctime
atof
atoi
atol
calloc
exit
fclose
feof
ferror
fflush
fgetc
fgets
fgetws
fopen
fopen_s
fprintf
fprintf_s
fputs
fread
free
freeaddrinfo
fseek
ftell
fwrite
getaddrinfo
getchar
getenv
getnameinfo
inet_ntop
inet_pton
isalnum
isdigit
isprint
isspace
isupper

FwpmFreeMemory0
FwpmGetAppIdFromFileName0
FwpmSubLayerAdd0
FwpmSubLayerDeleteByKey0
FwpmTransactionAbort0
FwpmTransactionBegin0
FwpmTransactionCommit0
GetAdaptersAddresses
GetAdaptersInfo
GetBestInterfaceEx
GetBestRoute
GetBestRoute2
GetBitmapBits
GetClassNameW
GetClientRect
GetComputerNameExW
GetCurrentDirectoryW
GetCurrentProcess
GetCurrentProcessId
GetCurrentThreadId
GetCursorPos
GetDC
GetDesktopWindow
GetDeviceCaps
GetDlgItem
GetEnvironmentVariableA
GetEnvironmentVariableW
GetExitCodeProcess
GetFileAttributesW
GetFileType
GetFocus
GetIfEntry
GetIfTable
GetIpAddrTable
GetIpForwardTable
GetIpForwardTable2
GetIpInterfaceTable
GetLastError

SendMessageW
SetActiveWindow
SetCursor
SetEvent
SetForegroundWindow
SetHandleInformation
SetIpForwardEntry
SetLastError
SetParent
SetRectEmpty
SetServiceStatus
SetTimer
SetTokenInformation
SetUnhandledExceptionFilter
SetWindowLongW
SetWindowPos
SetWindowTextW
SetupCloseFileQueue
SetupCloseInfFile
SetupCommitFileQueueA
SetupDefaultQueueCallbackA
SetupInitDefaultQueueCallback
SetupInstallFilesFromInfSectionW
SetupInstallFromInfSectionW
SetupInstallServicesFromInfSectionW
SetupOpenFileQueue
SetupOpenInfFileW
SetupTermDefaultQueueCallback
ShellExecuteW
Shell_NotifyIconW
Sleep
StartServiceA
StartServiceCtrlDispatcherA
StrStrIA
SystemParametersInfoW
TerminateProcess
TerminateThread
TextOutW

isxdigit
lstrcmpA
lstrlenA
lstrlenW
malloc
mbstowcs
mbstowcs_s
memchr
memcmp
memcpy
memcpy_s
memmove
memmove_s
memset
printf
qsort
raise
rand
realloc
rename
rewind
signal
sprintf
sprintf_s
srand
sscanf
sscanf_s
strcat
strcat_s
strchr
strcmp
strcpy
strcpy_s
strerror
strftime
strlen
strncat_s
strncmp

GetLocalTime

GetMenuItemCount

GetMenuItemInfoW

GetModuleFileNameA

GetModuleFileNameW

GetModuleHandleA

GetModuleHandleExW

GetModuleHandleW

GetNetworkParams

GetObjectW

GetParent

GetProcAddress

GetProcessHeap

GetProcessId

GetProcessImageFileNameW

GetProcessWindowStation

GetProfileType

GetProfilesDirectoryA

GetStdHandle

GetStockObject

GetSubMenu

GetSysColor

TrackMouseEvent

UnloadUserProfile

UnmapViewOfFile

UnregisterClassA

UpdateWindow

VerSetConditionMask

VerifyVersionInfoW

WSAAccept

WSACreateEvent

WSAEnumNetworkEvents

WSAEventSelect

WSASocketA

WTSEnumerateSessionsA

WTSFreeMemory

WTSGetActiveConsoleSessionId

WTSQuerySessionInformationA

WTSQueryUserToken

WTSRegisterSessionNotification

WaitForMultipleObjects

WaitForSingleObject

strncpy

strncpy_s

strnlen

strrchr

strstr

strtok_s

strtol

strtoul

swprintf_s

tolower

toupper

vfprintf

wcscat_s

wcschr

wcscmp

wcscpy_s

wcslen

wcsncat_s

wcsncmp

wcsncpy

wcsncpy_s

wcsnlen

wcsrchr

wcsstr

wcstok

wcstok_s

wcstombs_s

wprintf

wsprintfW

macOS APIs used by GlobalProtect

_AuthorizationCreate
_AuthorizationFree
_AuthorizationRightGet
_AuthorizationRightSet
_CFAllocatorCreate
_CFAllocatorGetContext
_CFArrayAppendValue
_CFArrayCreateMutable
_CFArrayGetCount
_CFArrayGetValueAtIndex
_CFAutorelease
_CFBooleanGetTypeID
_CFBooleanGetValue
_CFBundleCopyBundleURL
_CFBundleCopyResourceURL
_CFBundleCreate
_CFBundleGetMainBundle
_CFBundleGetValueForInfoDictionaryKey
_CFCopyDescription
_CFDataCreate
_CFDataCreateWithBytesNoCopy
_CFDataGetBytePtr
_CFDataGetBytes
_CFDataGetLength
_CFDictionaryAddValue
_CFDictionaryCreate
_CFDictionaryCreateMutable
_CFDictionaryGetCount
_CFDictionaryGetTypeID
_CFDictionaryGetValue
_CFDictionarySetValue
_CFEqual
_CFErrorGetCode
_CFGetRetainCount
_CFGetTypeID
_OBJC_METACLASS_$_WebView
_SCDynamicStoreCopyComputerName
_SCDynamicStoreCopyConsoleUser
_SCDynamicStoreCopyLocalHostName
_SCDynamicStoreCopyProxies
_SCDynamicStoreCopyValue
_SCDynamicStoreCreate
_SCDynamicStoreRemoveValue
_SCDynamicStoreSetValue
_SCError
_SCErrorString
_SCNetworkInterfaceCopyAll
_SCNetworkInterfaceGetBSDName
_SCNetworkInterfaceGetHardwareAddressString
_SCNetworkReachabilityCreateWithAddress
_SCNetworkReachabilityCreateWithName
_SCNetworkReachabilityGetFlags
_SecAccessControlCreateWithFlags
_SecAccessCreate
_SecCertificateCopyCommonName
_SecCertificateCopyData
_SecCertificateCopySubjectSummary
_SecCertificateCopyValues
_SecCertificateCreateWithData
_SecCertificateGetCLHandle
_SecCertificateGetData
_SecCertificateGetSubject
_SecCertificateGetTypeID
_SecCodeCheckValidityWithErrors
_SecCodeCopySelf
_SecCopyErrorMessageString
_SecDecryptTransformCreate
_SecIdentityCopyCertificate
_inet_addr
_inet_aton
_inet_ntoa
_inet_ntop
_inet_pton
_ioctl
_ivar_getName
_ivar_getOffset
_kCFAllocatorDefault
_kCFAllocatorMalloc
_kCFAllocatorNull
_kCFBooleanTrue
_kCFBundleVersionKey
_kCFCoreFoundationVersionNumber
_kCFPreferencesAnyHost
_kCFPreferencesCurrentUser
_kCFProxyHostNameKey
_kCFProxyPortNumberKey
_kCFRunLoopDefaultMode
_kCFTypeArrayCallBacks
_kCFTypeDictionaryKeyCallBacks
_kCFTypeDictionaryValueCallBacks
_kIOMasterPortDefault
_kSCPropInterfaceName
_kSCPropNetDNSSearchDomains
_kSCPropNetDNSSearchOrder
_kSCPropNetDNSServerAddresses
_kSCPropNetIPv4Addresses
_kSCPropNetIPv4Router
_kSCPropNetIPv4SubnetMasks
_kSCPropNetIPv6Addresses
_kSCPropNetIPv6PrefixLength
_kSCPropNetIPv6Router
_kSCPropNetOverridePrimary
_kSCPropNetProxiesHTTPSEnable

_CFMakeCollectable

_CFNetworkCopyProxiesForAutoConfigurationScript

_CFNumberCreate

_CFNumberGetTypeID

_CFNumberGetValue

_CFPreferencesCopyAppValue

_CFPreferencesCopyApplicationList

_CFPropertyListCreateData

_CFPropertyListCreateDeepCopy

_CFPropertyListCreateWithData

_CFPropertyListWriteToStream

_CFRelease

_CFRetain

_CFRunLoopAddSource

_CFRunLoopGetCurrent

_CFRunLoopGetMain

_CFRunLoopRun

_CFRunLoopStop

_CFSocketCreateRunLoopSource

_CFSocketCreateWithNative

_CFSocketGetSocketFlags

_CFSocketInvalidate

_CFSocketSetSocketFlags

_CFStringAppendCString

_CFStringCompare

_CFStringCreateArrayBySeparatingStrings

_CFStringCreateCopy

_CFStringCreateFromExternalRepresentation

_CFStringCreateMutable

_CFStringCreateMutableCopy

_CFStringCreateWithBytes

_CFStringCreateWithCString

_CFStringCreateWithFormat

_CFStringCreateWithFormatAndArguments

_CFStringGetCString

_CFStringGetCStringPtr

_CFStringGetLength

_SecIdentityCopyPrivateKey

_SecIdentityCreateWithCertificate

_SecIdentityGetTypeID

_SecIdentitySearchCopyNext

_SecIdentitySearchCreate

_SecItemAdd

_SecItemCopyMatching

_SecItemDelete

_SecItemImport

_SecItemUpdate

_SecKeychainAttributeInfoForItemID

_SecKeychainCopyDefault

_SecKeychainCopyDomainDefault

_SecKeychainCopyDomainSearchList

_SecKeychainCopySearchList

_SecKeychainFindInternetPassword

_SecKeychainFreeAttributeInfo

_SecKeychainGetPath

_SecKeychainGetStatus

_SecKeychainItemCopyAttributesAndData

_SecKeychainItemCopyFromPersistentReference

_SecKeychainItemCopyKeychain

_SecKeychainItemCreatePersistentReference

_SecKeychainItemDelete

_SecKeychainItemFreeAttributesAndData

_SecKeychainItemFreeContent

_SecKeychainItemImport

_SecKeychainOpen

_SecKeychainSearchCopyNext

_SecKeychainSearchCreateFromAttributes

_SecKeychainSetPreferenceDomain

_SecKeychainUnlock

_SecPKCS12Import

_SecPolicyCreateBasicX509

_kSCPropNetProxiesHTTPSPort

_kSCPropNetProxiesHTTPSProxy

_kSCPropNetProxiesProxyAutoConfigEnable

_kSCPropNetProxiesProxyAutoConfigURLString

_kSecAttrAccess

_kSecAttrAccessControl

_kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly

_kSecAttrAccount

_kSecAttrIsExtractable

_kSecAttrIsSensitive

_kSecAttrIssuer

_kSecAttrLabel

_kSecAttrService

_kSecAttrSubject

_kSecClass

_kSecClassCertificate

_kSecClassGenericPassword

_kSecClassIdentity

_kSecImportExportPassphrase

_kSecImportItemCertChain

_kSecImportItemIdentity

_kSecMatchLimit

_kSecMatchLimitAll

_kSecMatchLimitOne

_kSecMatchSearchList

_kSecOIDExtendedKeyUsage

_kSecPropertyTypeError

_kSecPropertyTypeTitle

_kSecReturnAttributes

_kSecReturnData

_kSecReturnPersistentRef

_kSecReturnRef

_kSecTransformInputAttributeName

_kSecUseKeychain

_kSecUseOperationPrompt

_kSecValueData

_CFStringGetMaximumSizeForEncoding

_CFStringGetTypeID

_CFURLCreateCopyAppendingPathComponent

_CFURLCreateDataAndPropertiesFromResource

_CFURLCreateFromFileSystemRepresentation

_CFURLCreateWithFileSystemPath

_CFURLCreateWithString

_CFUUIDCreate

_CFUUIDCreateString

_CFWriteStreamClose

_CFWriteStreamCreateWithFile

_CFWriteStreamOpen

_CGRectGetHeight

_CGRectGetWidth

_CGSessionCopyCurrentDictionary

_CGSizeZero

_CSSMOID_APPLE_TP_REVOCATION_CRL

_CSSMOID_APPLE_TP_REVOCATION_OCSP

_CSSMOID_APPLE_X509_BASIC

_CSSMOID_ClientAuth

_CSSMOID_CommonName

_CSSMOID_ExtendedKeyUsage

_CSSMOID_ExtendedKeyUsageAny

_CSSMOID_KeyUsage

_CSSMOID_ServerAuth

_CSSMOID_X509V1IssuerName

_CSSMOID_X509V1SubjectName

_CSSM_CL_CertGetAllFields

_CSSM_CL_FreeFields

_DNSServiceProcessResult

_DNSServiceQueryRecord

_DNSServiceRefDeallocate

_DNSServiceRefSockFD

_Gestalt

_IOIteratorNext

_SecPolicyCreateRevocation

_SecPolicyCreateSSL

_SecPolicySearchCopyNext

_SecPolicySearchCreate

_SecPolicySetValue

_SecRandomCopyBytes

_SecRequirementCreateWithString

_SecStaticCodeCheckValidityWithErrors

_SecStaticCodeCreateWithPath

_SecTransformExecute

_SecTransformSetAttribute

_SecTrustCopyProperties

_SecTrustCopyResult

_SecTrustCreateWithCertificates

_SecTrustEvaluate

_SecTrustGetCertificateAtIndex

_SecTrustGetCertificateCount

_SecTrustGetCssmResultCode

_SecTrustGetResult

_SecTrustSetAnchorCertificates

_SecTrustSetAnchorCertificatesOnly

_SecTrustSetKeychains

_SecTrustSetNetworkFetchAllowed

_SecTrustSetParameters

_SecTrustSettingsSetTrustSettings

_SecTrustedApplicationCreateFromPath

__Block_copy

__Block_object_assign

__Block_object_dispose

__DefaultRuneLocale

__NSConcreteGlobalBlock

__NSConcreteStackBlock

__NSDictionaryOfVariableBindings

__Unwind_Resume

___CFConstantStringClassReference

___assert_rtn

___bzero

_kSecValuePersistentRef

_kSecValueRef

_kill

_link_ntoa

_listen

_localtime

_localtime_r

_lseek

_lstat$INODE64

_mach_absolute_time

_mach_timebase_info

_malloc

_malloc_size

_memchr

_memcmp

_memcpy

_memmove

_memset

_method_setImplementation

_mkdir

_mktime

_nanosleep

_objc_alloc

_objc_allocateClassPair

_objc_autorelease

_objc_autoreleasePoolPop

_objc_autoreleasePoolPush

_objc_autoreleaseReturnValue

_objc_begin_catch

_objc_constructInstance

_objc_copyClassNamesForImage

_objc_destroyWeak

_objc_end_catch

_objc_enumerationMutation

_objc_getClass

_objc_getMetaClass

_objc_getProperty

_objc_getProtocol

_IOObjectRelease

_IORegistryEntryCreateCFProperty

_IORegistryEntryGetParentEntry

_IOServiceGetMatchingService

_IOServiceGetMatchingServices

_IOServiceMatching

_KextManagerCopyLoadedKextInfo

_KextManagerCreateURLForBundleIdentifier

_KextManagerLoadKextWithURL

_KextManagerUnloadKextWithIdentifier

_NSApp

_NSAppearanceNameAqua

_NSApplicationDidChangeScreenParametersNotification

_NSApplicationMain

_NSBeep

_NSCharacterEncodingDocumentAttribute

_NSContainsRect

_NSDefaultRunLoopMode

_NSDocumentTypeDocumentAttribute

_NSEdgeInsetsZero

_NSFileGroupOwnerAccountID

_NSFileGroupOwnerAccountName

_NSFileOwnerAccountID

_NSFileOwnerAccountName

_NSFilePosixPermissions

_NSFontAttributeName

_NSFontWeightBold

_NSFontWeightLight

_NSFontWeightMedium

_NSForegroundColorAttributeName

_NSFoundationVersionNumber

_NSHTMLTextDocumentType

_NSHomeDirectory

_NSInsetRect

_NSLocalizedDescriptionKey

_NSLog

_NSOffsetRect

___cxa_allocate_exception

___cxa_atexit

___cxa_begin_catch

___cxa_call_unexpected

___cxa_end_catch

___cxa_free_exception

___cxa_guard_abort

___cxa_guard_acquire

___cxa_guard_release

___cxa_pure_virtual

___cxa_throw

___error

___gxx_personality_v0

___maskrune

___memcpy_chk

___memmove_chk

___memset_chk

___objc_personality_v0

___sprintf_chk

___stack_chk_fail

___stack_chk_guard

___stderrp

___stdinp

___stdoutp

___strcat_chk

___tolower

___toupper

__dispatch_main_q

__dispatch_source_type_timer

__dispatch_source_type_vnode

__dyld_register_func_for_add_image

__exit

__objc_empty_cache

__objc_empty_vtable

_abort

_accept

_access

_arc4random

_objc_getRequiredClass

_objc_initializeClassPair

_objc_loadClassref

_objc_loadWeakRetained

_objc_lookUpClass

_objc_msgSend

_objc_msgSendSuper2

_objc_msgSendSuper2_stret

_objc_msgSend_stret

_objc_readClassPair

_objc_registerClassPair

_objc_release

_objc_retain

_objc_retainAutorelease

_objc_retainAutoreleaseReturnValue

_objc_retainAutoreleasedReturnValue

_objc_setProperty_atomic

_objc_setProperty_atomic_copy

_objc_setProperty_nonatomic

_objc_setProperty_nonatomic_copy

_objc_storeStrong

_objc_storeWeak

_objc_sync_enter

_objc_sync_exit

_object_getClass

_object_getIndexedIvars

_object_getIvar

_object_setIvar

_open

_opendir$INODE64

_pclose

_popen

_posix_spawn

_posix_spawn_file_actions_addinherit_np

_posix_spawn_file_actions_destroy

_posix_spawn_file_actions_init

_posix_spawnattr_destroy

_posix_spawnattr_init

_NSParagraphStyleAttributeName

_NSPointInRect

_NSRectFill

_NSRunAlertPanel

_NSSearchPathForDirectoriesInDomains

_NSStringFromSelector

_NSTemporaryDirectory

_NSURLAuthenticationMethodClientCertificate

_NSURLAuthenticationMethodServerTrust

_NSUnderlineStyleAttributeName

_NSWindowDidMoveNotification

_NSWindowDidResizeNotification

_NSWorkspaceDidWakeNotification

_NSWorkspaceSessionDidBecomeActiveNotification

_NSWorkspaceSessionDidResignActiveNotification

_NSWorkspaceWillPowerOffNotification

_NSWorkspaceWillSleepNotification

_NSZeroRect

_OBJC_CLASS_$_CATextLayer

_OBJC_CLASS_$_CWInterface

_OBJC_CLASS_$_LAContext

_OBJC_CLASS_$_NSAlert

_OBJC_CLASS_$_NSApplication

_OBJC_CLASS_$_NSArray

_OBJC_CLASS_$_NSAssertionHandler

_OBJC_CLASS_$_NSAttributedString

_OBJC_CLASS_$_NSAutoreleasePool

_OBJC_CLASS_$_NSBezierPath

_OBJC_CLASS_$_NSBundle

_OBJC_CLASS_$_NSButton

_OBJC_CLASS_$_NSCharacterSet

_OBJC_CLASS_$_NSColor

_OBJC_CLASS_$_NSData

_OBJC_CLASS_$_NSDate

_OBJC_CLASS_$_NSDateFormatter

_OBJC_CLASS_$_NSDictionary

_arc4random_buf

_asctime

_asprintf

_atof

_atoi

_atol

_atoll

_backtrace

_backtrace_symbols

_basename

_bind

_bzero

_calloc

_ceil

_chmod

_chown

_class_addMethod

_class_addProperty

_class_addProtocol

_class_copyPropertyList

_class_getInstanceMethod

_class_getInstanceSize

_class_getInstanceVariable

_class_getIvarLayout

_class_getName

_class_getSuperclass

_class_isMetaClass

_class_replaceMethod

_class_respondsToSelector

_clock_gettime

_close

_closedir

_connect

_dirname

_dispatch_after

_dispatch_async

_dispatch_get_global_queue

_dispatch_group_async

_posix_spawnattr_setflags

_pow

_printf

_proc_listallpids

_proc_pidpath

_property_copyAttributeList

_property_getName

_protocol_getMethodDescription

_protocol_getName

_pthread_attr_destroy

_pthread_attr_init

_pthread_cancel

_pthread_cond_destroy

_pthread_cond_init

_pthread_cond_signal

_pthread_cond_timedwait

_pthread_cond_wait

_pthread_create

_pthread_detach

_pthread_exit

_pthread_join

_pthread_mach_thread_np

_pthread_mutex_destroy

_pthread_mutex_init

_pthread_mutex_lock

_pthread_mutex_unlock

_pthread_mutexattr_destroy

_pthread_mutexattr_init

_pthread_mutexattr_settype

_pthread_self

_puts

_qsort

_rand

_read

_readdir$INODE64

_realloc

_recv

_recvfrom

_OBJC_CLASS_$_NSError

_OBJC_CLASS_$_NSEvent

_OBJC_CLASS_$_NSFileManager

_OBJC_CLASS_$_NSFont

_OBJC_CLASS_$_NSHTTPCookieStorage

_OBJC_CLASS_$_NSImage

_OBJC_CLASS_$_NSImageView

_OBJC_CLASS_$_NSInvocation

_OBJC_CLASS_$_NSJSONSerialization

_OBJC_CLASS_$_NSLayoutConstraint

_OBJC_CLASS_$_NSMenu

_OBJC_CLASS_$_NSMenuItem

_OBJC_CLASS_$_NSMutableArray

_OBJC_CLASS_$_NSMutableAttributedString

_OBJC_CLASS_$_NSMutableData

_OBJC_CLASS_$_NSMutableDictionary

_OBJC_CLASS_$_NSMutableOrderedSet

_OBJC_CLASS_$_NSMutableParagraphStyle

_OBJC_CLASS_$_NSMutableSet

_OBJC_CLASS_$_NSMutableString

_OBJC_CLASS_$_NSMutableURLRequest

_OBJC_CLASS_$_NSNotificationCenter

_OBJC_CLASS_$_NSNull

_OBJC_CLASS_$_NSNumber

_OBJC_CLASS_$_NSNumberFormatter

_OBJC_CLASS_$_NSObject

_OBJC_CLASS_$_NSOperationQueue

_OBJC_CLASS_$_NSOutlineView

_OBJC_CLASS_$_NSPanel

_OBJC_CLASS_$_NSPipe

_OBJC_CLASS_$_NSPopUpButton

_OBJC_CLASS_$_NSPredicate

_OBJC_CLASS_$_NSProcessInfo

_OBJC_CLASS_$_NSProgressIndicator

_OBJC_CLASS_$_NSRunLoop

_OBJC_CLASS_$_NSScanner

_OBJC_CLASS_$_NSScreen

_dispatch_group_create

_dispatch_group_enter

_dispatch_group_leave

_dispatch_group_wait

_dispatch_once

_dispatch_queue_create

_dispatch_queue_get_label

_dispatch_release

_dispatch_resume

_dispatch_retain

_dispatch_source_cancel

_dispatch_source_create

_dispatch_source_set_cancel_handler

_dispatch_source_set_event_handler

_dispatch_source_set_timer

_dispatch_time

_dladdr

_dlclose

_dlerror

_dlopen

_dlsym

_endutxent

_environ

_execl

_exit

_fchown

_fclose

_fcntl

_fcopyfile

_feof

_ferror

_fflush

_fgets

_fileno

_floor

_fopen

_fork

_fprintf

_remove

_rename

_res_9_getservers

_res_9_ndestroy

_res_9_ninit

_rewind

_round

_roundf

_sel_getUid

_select$1050

_send

_setenv

_seteuid

_setlogin

_setreuid

_setsockopt

_setuid

_setutxent

_setvbuf

_shmat

_shmctl

_shmget

_shutdown

_sigaction

_signal

_sleep

_snprintf

_socket

_socketpair

_sprintf

_srand

_sscanf

_stat$INODE64

_stpncpy

_strcasecmp

_strcasestr

_strchr

_strcmp

_OBJC_CLASS_$_NSScrollView

_OBJC_CLASS_$_NSSecureTextField

_OBJC_CLASS_$_NSStackView

_OBJC_CLASS_$_NSStatusBar

_OBJC_CLASS_$_NSString

_OBJC_CLASS_$_NSTabView

_OBJC_CLASS_$_NSTabViewItem

_OBJC_CLASS_$_NSTableColumn

_OBJC_CLASS_$_NSTableView

_OBJC_CLASS_$_NSTask

_OBJC_CLASS_$_NSTextField

_OBJC_CLASS_$_NSThread

_OBJC_CLASS_$_NSTimer

_OBJC_CLASS_$_NSTrackingArea

_OBJC_CLASS_$_NSURL

_OBJC_CLASS_$_NSURLCache

_OBJC_CLASS_$_NSURLComponents

_OBJC_CLASS_$_NSURLConnection

_OBJC_CLASS_$_NSURLCredential

_OBJC_CLASS_$_NSURLRequest

_OBJC_CLASS_$_NSURLSession

_OBJC_CLASS_$_NSURLSessionConfiguration

_OBJC_CLASS_$_NSUserDefaults

_OBJC_CLASS_$_NSView

_OBJC_CLASS_$_NSViewController

_OBJC_CLASS_$_NSWindow

_OBJC_CLASS_$_NSWindowController

_OBJC_CLASS_$_NSWorkspace

_OBJC_CLASS_$_NSXMLDocument

_OBJC_CLASS_$_NSXMLElement

_OBJC_CLASS_$_NSXMLNode

_OBJC_CLASS_$_NSXMLParser

_OBJC_CLASS_$_SFCertificatePanel

_OBJC_CLASS_$_SFChooseIdentityPanel

_OBJC_CLASS_$_WebView

_OBJC_EHTYPE_$_NSException

_OBJC_METACLASS_$_NSAlert

_fputc

_fputs

_fread

_free

_freeaddrinfo

_freeifaddrs

_freopen

_fscanf

_fseek

_fstat$INODE64

_ftell

_fwrite

_gai_strerror

_getaddrinfo

_getcwd

_getegid

_getenv

_geteuid

_getgid

_getgrnam_r

_gethostbyname

_gethostbyname2

_getifaddrs

_getlogin

_getnameinfo

_getpid

_getppid

_getpwnam

_getpwuid

_getpwuid_r

_getservbyname

_getsockname

_getsockopt

_gettimeofday

_getuid

_getutxent

_gmtime

_gmtime_r

_strcpy

_strdup

_strerror

_strerror_r

_strftime

_strlen

_strncasecmp

_strncat

_strncmp

_strncpy

_strnlen

_strnstr

_strptime

_strrchr

_strstr

_strtof

_strtok_r

_strtol

_strtoul

_strtoull

_symlink

_syscall

_sysconf

_sysctl

_sysctlbyname

_syslog

_system

_tcgetattr

_tcsetattr

_time

_unlink

_unsetenv

_usleep

_utimes

_vasprintf

_vfprintf

_vsnprintf

_waitpid

_OBJC_METACLASS_$_NSButton

_OBJC_METACLASS_$_NSMenuItem

_OBJC_METACLASS_$_NSObject

_OBJC_METACLASS_$_NSPanel

_OBJC_METACLASS_$_NSView

_OBJC_METACLASS_$_NSViewController

_OBJC_METACLASS_$_NSWindowController

_h_errno

_hash_create

_hash_search

_hstrerror

_if_indextoname

_if_nametoindex

_in6addr_any

_write

_xar_close

_xar_extract_tobuffersz

_xar_file_first

_xar_file_next

_xar_get_path

_xar_iter_free

_xar_iter_new

_xar_open

dyld_stub_binder

operator delete(void*)

operator delete[](void*)

operator new(unsigned long)

operator new[](unsigned long)