Assurance Activities Report
for
Palo Alto Networks GlobalProtect App 6

Version 1.0
28 August 2023

Evaluated By:

Leidos Inc.
https://www.leidos.com/civil/commercial-cyber/product-compliance
Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive
Columbia, MD 21046

Prepared for:
National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:
Palo Alto Networks, Inc.
3000 Tannery Way
Santa Clara, CA 95054

The TOE Evaluation was Sponsored by:
Palo Alto Networks, Inc.
3000 Tannery Way
Santa Clara, CA 95054

Evaluation Personnel:
Anthony Apted
Pascal Patin
Armin Najafabadi

Common Criteria Version:

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1, Revision 5, April 2017.
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.

Common Evaluation Methodology Version:

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 5, April 2017.

Protection Profiles:

- Protection Profile for Application Software, Version 1.4, 7 October 2021
- Functional Package for Transport Layer Security (TLS), Version 1.1, 1 March 2019

Revision History

Version	Date	Description
0.1	23 June 2023	Initial draft
1.0	28 August 2023	Final version for check-out

Contents

1. Introduction

This document presents the results of performing assurance activities associated with the Palo Alto Networks GlobalProtect App 6 evaluation. This report contains sections documenting the performance of evaluation activities associated with each of the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) as specified in the following documents:

- *Protection Profile for Application Software*, Version 1.4, 7 October 2021 [PP_APP_v1.4]
- *Functional Package for Transport Layer Security (TLS)*, Version 1.1, 1 March 2019 [PKG_TLS_v1.1]

Note that, in accordance with NIAP Policy Letter #5, all cryptography in the TOE for which NIST provides validation testing of FIPS-approved and NIST-recommended cryptographic algorithms and their individual components must be NIST validated. The CCTL will verify that the claimed NIST validation complies with the NIAP-approved PP requirements the TOE claims to satisfy. The CCTL verification of the NIST validation will constitute performance of the associated assurance activity. As such, Test assurance activities associated with functional requirements within the scope of Policy Letter #5 are performed by verification of the relevant CAVP certification and not through performance of any testing as specified in the claimed PP documents.

1.1 Technical Decisions

1.1.1 Protection Profile for Application Software

This subsection lists the Technical Decisions that have been issued by NIAP against [PP_APP_v1.4], along with rationale as to their applicability or otherwise to this evaluation.

TD0756: Update for platform-provided full disk encryption

- This TD has been applied to this evaluation.

TD0743: FTP_DIT_EXT.1.1 Selection exclusivity

- This TD has been applied to this evaluation.

TD0736: Number of elements for iterations of FCS_HTTPS_EXT.1

- N/A – the TOE does not claim this SFR.

TD0719: ECD for PP APP V1.3 and 1.4

- This TD has been applied to this evaluation.

TD0717: Format changes for PP_APP_V1.4

- This TD has been applied to this evaluation.

TD0669: FIA_X509_EXT.1 Test 4 Interpretation

- This TD has been applied to this evaluation.

TD0664: Testing activity for FPT_TUD_EXT.2.2

- This TD has been applied to this evaluation.

TD0650: Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4

- N/A – the TOE does not claim VPN client functionality.

TD0628: Addition of Container Image to Package Format

- This TD has been applied to this evaluation.

TD0624: Addition of DataStore for Storing and Setting Configuration Options

- This TD has been applied to this evaluation.

1.1.2 Functional Package for Transport Layer Security (TLS)

This subsection lists the Technical Decisions that have been issued by NIAP against [PKG_TLS_V1.1], along with rationale as to their applicability or otherwise to this evaluation.

TD0770: TLSS.2 connection with no client cert

- N/A – the TOE does not claim this SFR.

TD0739: PKG_TLS_V1.1 has 2 different publication dates

- This TD has been applied to this evaluation.

TD0726: Corrections to (D)TLSS SFRs in TLS 1.1 FP

- N/A – the TOE does not claim these SFRs.

TD0588: Session Resumption Support in TLS package

- N/A – the TOE does not claim the affected SFRs.

TD0513: CA Certificate loading

- This TD has been applied to this evaluation.

TD0499: Testing with pinned certificates

- This TD has been applied to this evaluation.

TD0469: Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1

- N/A – the TOE does not claim this SFR.

TD0442: Updated TLS Ciphersuites for TLS Package

- This TD has been applied to this evaluation.

1.2 SAR Evaluation

The following Security Assurance Requirements (SARs) were evaluated during the evaluation of the TOE:

SAR	Verdict
ASE_CCL.1	Pass
ASE_ECD.1	Pass
ASE_INT.1	Pass
ASE_OBJ.1	Pass
ASE_REQ.1	Pass
ASE_TSS.1	Pass
ADV_FSP.1	Pass
AGD_OPE.1	Pass

SAR	Verdict
AGD_PRE.1	Pass
ALC_CMC.1	Pass
ALC_CMS.1	Pass
ALC_TSU_EXT.1	Pass
ATE_IND.1	Pass
AVA_VAN.1	Pass

The evaluation work units are listed in the proprietary ETR. The evaluators note per the PP evaluation activities that many of the SARs were successfully evaluated through completion of the associated evaluation activities present in the claimed PP and PP-Modules.

1.3 References

[ST] *Palo Alto Networks GlobalProtect App 6 Security Target*, version 1.0, 26 July 2023

[CCECG] *Palo Alto Networks Common Criteria Evaluated Configuration Guide (CCECG) GlobalProtect 6 App*, 28 June 2023

2. Security Functional Requirement Evaluation Activities

This section describes the evaluation activities associated with the SFRs defined in the ST and the results of those activities as performed by the evaluation team. The evaluation activities are derived from [PP_APP_v1.4] and [PKG_TLS_V1.1]. NIAP Technical Decisions have been applied and are identified as appropriate.

2.1 Cryptographic Support (FCS)

The following table lists the cryptographic functions supported by the TOE and associated SFRs, the specific algorithms that are claimed for these functions, and the relevant CAVP certificate validation lists and certificate numbers for each.

Functions	Standards	Certificates
FCS_CKM.1/AK Cryptographic Asymmetric Key Generation		
ECC key pair generation (NIST curves P-256, P-384, P-521)	FIPS PUB 186-4	A2999: ECDSA KeyGen (FIPS186-4)
FCS_CKM.2 Cryptographic Key Establishment		
ECDSA based key establishment	NIST SP 800-56A	A2999: KAS-ECC-SSC Sp800-56Ar3
FCS_COP.1/SKC Cryptographic Operation – Encryption/Decryption		
AES-CBC (128, 256 bits)	CBC as defined in NIST SP 800-38A	A2999: AES-CBC
AES-GCM (128, 256 bits)	GCM as defined in NIST SP 800-38D	A2999: AES-GCM
FCS_COP.1/Hash Cryptographic Operation – Hashing		
SHA-1, SHA-256, SHA-384 (digest sizes 160, 256, and 384 bits)	FIPS PUB 180-4	A2999: SHA-1 A2999: SHA2-256 A2999: SHA2-384
FCS_COP.1/Sig Cryptographic Operation – Signing		
RSA (2048-bit or greater)	FIPS PUB 186-4, Section 5	A2999: RSA SigGen (FIPS 186-4) A2999: RSA SigVer (FIPS 186-4)
ECDSA with NIST curves P-256, P-384, and P-521	FIPS PUB 186-4, Section 6	A2999: ECDSA SigGen (FIPS 186-4) A2999: ECDSA SigVer (FIPS 186-4)
FCS_COP.1/KeyedHash Cryptographic Operation – Keyed Hash Message Authentication		
HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-384	FIPS PUB 198-1 FIPS PUB 180-4	A2999: HMAC-SHA1-1 A2999: HMAC-SHA2-256 A2999: HMAC-SHA2-384
FCS_RBG_EXT.2 Random Bit Generation from Application		
CTR_DRBG (AES), 256 bits	NIST SP 800-90A NIST SP 800-57	A2999: Counter DRBG

2.1.1 FCS_CKM_EXT.1 Cryptographic Asymmetric Key Generation

2.1.1.1 TSS Evaluation Activity

The evaluator shall inspect the application and its developer documentation to determine if the application needs asymmetric key generation services. If not, the evaluator shall verify the "***generate no asymmetric cryptographic keys***" selection is present in the ST. Otherwise, the evaluation activities shall be performed as stated in the selection-based requirements.

The TOE requires asymmetric key generation services to support Diffie-Hellman elliptic curve key establishment functionality used in TLS connections. The TOE includes its own cryptographic module that implements its required asymmetric key generation functionality. The ST selects "implement asymmetric key generation" in FCS_CKM_EXT.1.1.

2.1.1.2 Guidance Evaluation Activity

None.

2.1.1.3 Test Evaluation Activity

None.

2.1.2 FCS_CKM.1/AK Cryptographic Asymmetric Key Generation

2.1.2.1 TSS Evaluation Activity

The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

The statement of FCS_CKM.1/AK in section 5.2.1 of [ST] ("Cryptographic Support (FCS)") specifies the TOE implements functionality to generate asymmetric cryptographic keys using Elliptic Curve Cryptography (ECC) schemes using NIST curves P-256, P-384, and P-521.

Section 6.1 of [ST] ("Cryptographic Support") states the TOE utilizes ECDHE (ephemeral Elliptic Curve Diffie-Hellman) for TLS key establishment with NIST curves P-256, P-384, and P-521.

If the application "invokes platform-provided functionality for asymmetric key generation," then the evaluator shall examine the TSS to verify that it describes how the key generation functionality is invoked.

The ST does not select "invokes platform-provided functionality for asymmetric key generation", so this activity is not applicable.

2.1.2.2 Guidance Evaluation Activity

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all uses defined in this PP.

Section "Software Download and Installation", subsection "Installation" of [CCECG] provides instructions to place the TOE in its evaluated configuration. This includes enabling FIPS-CC mode, which restricts the TOE to using only the cipher suites claimed in the ST for TLS connections and to generating asymmetric keys as part of key exchange using NIST curves P-256, P-384, and P-521 only.

2.1.2.3 Test Evaluation Activity

If the application "implements asymmetric key generation," then the following test activities shall be carried out.

Evaluation Activity Note: The following tests may require the developer to provide access to a developer environment that provides the evaluator with tools that are typically available to end-users of the application.

Performed in accordance with NIAP Policy Letter #5.

Section 6.1 of [ST] ("Cryptographic Support"), Table 4 ("Cryptographic Functions and CAVP Certificates") identifies the CAVP certifications verifying validation for asymmetric key generation, as follows.

Algorithm	Tested Capabilities	Certificates
ECC schemes using NIST curves P-256, P-384, and P-521 that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	Curve: P-256, P-384, P-521 Secret Generation Mode: Testing Candidates	CAVP #A2999 (ECDSA)

2.1.3 FCS_CKM.2 Cryptographic Key Establishment

2.1.3.1 TSS Evaluation Activity

Modified in accordance with TD0717.

The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1/AK. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

FCS_CKM.1.1/AK specifies asymmetric key generation using ECC schemes. FCS_CKM.2.1 specifies the use of elliptical curve-based key establishment schemes, consistent with the key generation schemes specified in FCS_CKM.1.1/AK.

2.1.3.2 Guidance Evaluation Activity

The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).

Section "Software Download and Installation", subsection "Installation" of [CCECG] provides instructions to place the TOE in its evaluated configuration. This includes enabling FIPS-CC mode, which restricts the TOE to using only the cipher suites claimed in the ST for TLS connections. The TLS cipher suites claimed for the TOE in its evaluated configuration all use elliptic curve-based key establishment schemes.

2.1.3.3 Test Evaluation Activity

***Evaluation Activity Note**: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.*

Key Establishment Schemes

The evaluator shall verify the implementation of the key establishment schemes supported by the TOE using the applicable tests below.

SP800-56A Key Establishment Schemes Performed in accordance with NIAP Policy Letter #5.

Section 6.1 of [ST] ("Cryptographic Support"), Table 4 ("Cryptographic Functions and CAVP Certificates") identifies the CAVP certifications verifying validation for cryptographic key establishment, as follows.

Algorithm	Tested Capabilities	Certificates
Elliptic curve-based key establishment schemes that meet NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	Domain Parameter Generation Methods: P-256, P-384, P-521 Scheme: ephemeralUnified: KAS Role: initiator, responder	CAVP #A2999 (KAS-ECC-SSC Sp800-56Ar3)

2.1.4 FCS_COP.1/SKC Cryptographic Operation - Encryption/Decryption

2.1.4.1 TSS Evaluation Activity

None.

2.1.4.2 Guidance Evaluation Activity

The evaluator checks the AGD documents to determine that any configuration that is required to be done to configure the functionality for the required modes and key sizes is present.

Section "Software Download and Installation", subsection "Installation" of [CCECG] provides instructions to place the TOE in its evaluated configuration. This includes enabling FIPS-CC mode, which restricts the TOE to using only the cipher suites claimed in the ST for TLS connections. This in turn restricts the modes and key sizes used to those specified in the requirement.

2.1.4.3 Test Evaluation Activity

Performed in accordance with NIAP Policy Letter #5.

Section 6.1 of [ST] ("Cryptographic Support"), Table 4 ("Cryptographic Functions and CAVP Certificates") identifies the CAVP certifications verifying validation for symmetric key encryption and decryption, as follows.

Algorithm	Tested Capabilities	Certificates
AES-CBC as defined in NIST SP 800-38A	Direction: Decrypt, Encrypt Key Length: 128, 256	CAVP #A2999 (AES-CBC)

Algorithm	Tested Capabilities	Certificates
AES-GCM as defined in NIST SP 800-38D	Direction: Decrypt, Encrypt IV Generation: Internal IV Generation Mode: 8.2.1 Key Length: 128, 256 Tag Length: 64, 96, 104, 112, 120, 128 IV Length: 96 Payload Length: 8-65536 Increment 8 AAD Length: 0-65536 Increment 8	CAVP #A2999 (AES-GCM)

2.1.5 FCS_COP.1/Hash Cryptographic Operation - Hashing

2.1.5.1 TSS Evaluation Activity

The evaluator shall check that the association of the hash function with other application cryptographic functions (for example, the digital signature verification function) is documented in the TSS.

Section 6.1 of [ST] ("Cryptographic Support") states the TOE uses hash functions 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 purposes. The TOE uses SHA-256 and SHA-384 hashing as part of generating digital signatures. The TOE uses SHA-1 as part of the software integrity power-up test.

2.1.5.2 Guidance Evaluation Activity

None.

2.1.5.3 Test Evaluation Activity

Performed in accordance with NIAP Policy Letter #5.

Section 6.1 of [ST] ("Cryptographic Support"), Table 4 ("Cryptographic Functions and CAVP Certificates") identifies the CAVP certifications verifying validation for cryptographic hashing, as follows.

Algorithm	Tested Capabilities	Certificates
SHA-1 that meets FIPS Pub 180-4.	Message Length: 0-65536 Increment 8	CAVP #A2999 (SHA-1)
SHA-256 that meets FIPS Pub 180-4.	Message Length: 0-65536 Increment 8	CAVP #A2999 (SHA2-256)
SHA-384 that meets FIPS Pub 180-4.	Message Length: 0-65536 Increment 8	CAVP #A2999 (SHA2-384)

2.1.6 FCS_COP.1/Sig Cryptographic Operation - Signing

2.1.6.1 TSS Evaluation Activity

None.

2.1.6.2 Guidance Evaluation Activity

None.

2.1.6.3 Test Evaluation Activity

Performed in accordance with NIAP Policy Letter #5.

Section 6.1 of [ST] ("Cryptographic Support"), Table 4 ("Cryptographic Functions and CAVP Certificates") identifies the CAVP certifications verifying validation for cryptographic signature generation and verification, as follows.

Algorithm	Tested Capabilities	Certificates
RSA schemes using cryptographic key sizes of 2048-bit or greater that meet FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.	**RSA Signature Generation:** Signature Type: PKCS 1.5 Properties: Modulo: 2048 Hash Pair: Hash Algorithm: SHA2-256 Hash Pair: Hash Algorithm: SHA2-384 Hash Pair: Hash Algorithm: SHA2-512 Properties: Modulo: 3072 Hash Pair: Hash Algorithm: SHA2-256 Hash Pair: Hash Algorithm: SHA2-384 Hash Pair: Hash Algorithm: SHA2-512 Properties: Modulo: 4096 Hash Pair: Hash Algorithm: SHA2-256 Hash Pair: Hash Algorithm: SHA2-384 Hash Pair: Hash Algorithm: SHA2-512	CAVP #A2999 (RSA SigGen (FIPS186-4))

Algorithm	Tested Capabilities	Certificates
	Signature Type: PKCSPSS Properties: Modulo: 2048 Hash Pair: Hash Algorithm: SHA2-256 Salt Length: 32 Hash Pair: Hash Algorithm: SHA2-384 Salt Length: 48 Hash Pair: Hash Algorithm: SHA2-512 Salt Length: 64 Properties: Modulo: 3072 Hash Pair: Hash Algorithm: SHA2-256 Salt Length: 32 Hash Pair: Hash Algorithm: SHA2-384 Salt Length: 48 Hash Pair: Hash Algorithm: SHA2-512 Salt Length: 64 Properties: Modulo: 4096 Hash Pair: Hash Algorithm: SHA2-256 Salt Length: 32 Hash Pair: Hash Algorithm: SHA2-384 Salt Length: 48 Hash Pair: Hash Algorithm: SHA2-512 Salt Length: 64	

Algorithm	Tested Capabilities	Certificates
	RSA Signature Verification: Signature Type: PKCS 1.5 Properties: Modulo: 2048 Hash Pair: Hash Algorithm: SHA-1 Hash Pair: Hash Algorithm: SHA2-256 Hash Pair: Hash Algorithm: SHA2-384 Hash Pair: Hash Algorithm: SHA2-512 Properties: Modulo: 3072 Hash Pair: Hash Algorithm: SHA-1 Hash Pair: Hash Algorithm: SHA2-256 Hash Pair: Hash Algorithm: SHA2-384 Hash Pair: Hash Algorithm: SHA2-512 Properties: Modulo: 4096 Hash Pair: Hash Algorithm: SHA-1 Hash Pair: Hash Algorithm: SHA2-256 Hash Pair: Hash Algorithm: SHA2-384 Hash Pair: Hash Algorithm: SHA2-512	CAVP #A2999 (RSA SigVer (FIPS186-4))

Algorithm	Tested Capabilities	Certificates
	Signature Type: PKCSPSS Properties: Modulo: 2048 Hash Pair: Hash Algorithm: SHA2-256 Salt Length: 32 Hash Pair: Hash Algorithm: SHA2-384 Salt Length: 48 Hash Pair: Hash Algorithm: SHA2-512 Salt Length: 64 Properties: Modulo: 3072 Hash Pair: Hash Algorithm: SHA2-256 Salt Length: 32 Hash Pair: Hash Algorithm: SHA2-384 Salt Length: 48 Hash Pair: Hash Algorithm: SHA2-512 Salt Length: 64 Properties: Modulo: 4096 Hash Pair: Hash Algorithm: SHA2-256 Salt Length: 32 Hash Pair: Hash Algorithm: SHA2-384 Salt Length: 48 Hash Pair: Hash Algorithm: SHA2-512 Salt Length: 64 Public Exponent Mode: Fixed Fixed Public Exponent: 010001	
ECDSA schemes using NIST curves P-256, P-384 and P-521 that meet FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6.	**ECDSA Signature Generation:** Curve: P-256, P-384, P-521 Hash Algorithm: SHA2-256, SHA2-384, SHA2-512	CAVP #A2999 (ECDSA SigGen (FIPS186-4))

Algorithm	Tested Capabilities	Certificates
	ECDSA Signature Verification: Curve: P-256, P-384, P-521 Hash Algorithm: SHA-1, SHA2-256, SHA2-384, SHA2-512	CAVP #A2999 (ECDSA SigVer (FIPS186-4))

2.1.7 FCS_COP.1/KeyedHash Cryptographic Operation - Keyed-Hash Message Authentication

2.1.7.1 TSS Evaluation Activity

None.

2.1.7.2 Guidance Evaluation Activity

None.

2.1.7.3 Test Evaluation Activity

Performed in accordance with NIAP Policy Letter #5.

Section 6.1 of [ST] ("Cryptographic Support"), Table 4 ("Cryptographic Functions and CAVP Certificates") identifies the CAVP certifications verifying validation for keyed-hash message authentication, as follows.

Algorithm	Tested Capabilities	Certificates
HMAC-SHA-1 that meets FIPS Pub 198-1 and FIPS Pub 180-4.	MAC: 160 Key Length: 256-2048 Increment 8	CAVP #A2999 (HMAC-SHA-1)
HMAC-SHA-256 that meets FIPS Pub 198-1 and FIPS Pub 180-4.	MAC: 256 Key Length: 256-2048 Increment 8	CAVP #A2999 (HMAC-SHA2-256)
HMAC-SHA-384 that meets FIPS Pub 198-1 and FIPS Pub 180-4.	MAC: 384 Key Length: 256-2048 Increment 8	CAVP #A2999 (HMAC-SHA2-384)

2.1.8 FCS_RBG_EXT.1 Random Bit Generation Services

2.1.8.1 TSS Evaluation Activity

If ***use no DRBG functionality*** is selected, the evaluator shall inspect the application and its developer documentation and verify that the application needs no random bit generation services.

If ***implement DRBG functionality*** is selected, the evaluator shall ensure that additional FCS_RBG_EXT.2 elements are included in the ST.

If ***invoke platform-provided DRBG functionality*** is selected, the evaluator performs the following activities. The evaluator shall examine the TSS to confirm that it identifies all functions (as described by the SFRs included in the ST) that obtain random numbers from the platform RBG. The evaluator shall determine that for each of these functions, the TSS states which platform interface (API) is used to obtain the random numbers. The evaluator shall confirm that each of these interfaces corresponds to the acceptable interfaces listed for each platform below.

It should be noted that there is no expectation that the evaluators attempt to confirm that the APIs are being used "correctly" for the functions identified in the TSS; the activity is to list the used APIs and then do an existence check via decompilation.

The statement of FCS_RBG_EXT.1 in section 5.2.1 of [ST] ("Cryptographic Support (FCS)") selects "implement DRBG functionality". The evaluator confirmed the ST included FCS_RBG_EXT.2.

2.1.8.2 Guidance Evaluation Activity

None.

2.1.8.3 Test Evaluation Activity

If ***invoke platform-provided DRBG functionality*** is selected, the following tests shall be performed:

The evaluator shall decompile the application binary using a decompiler suitable for the application (TOE). The evaluator shall search the output of the decompiler to determine that, for each API listed in the TSS, that API appears in the output. If the representation of the API does not correspond directly to the strings in the following list, the evaluator shall provide a mapping from the decompiled text to its corresponding API, with a description of why the API text does not directly correspond to the decompiled text and justification that the decompiled text corresponds to the associated API.

If invocation of platform-provided functionality is achieved in another way, the evaluator shall ensure the TSS describes how this is carried out, and how it is equivalent to the methods listed here (e.g. higher-level API invokes identical low-level API).

The ST does not select "invoke platform-provided DRBG functionality", so this test activity is not applicable.

2.1.9 FCS_RBG_EXT.2 Random Bit Generation from Application

2.1.9.1 TSS Evaluation Activity

Documentation shall be produced - and the evaluator shall perform the activities - in accordance with Appendix C - Entropy Documentation and Assessment and the Clarification to the Entropy Documentation and Assessment Annex.

The vendor produced a proprietary Entropy Analysis Report (EAR) that the evaluators determined was suitable to meet the requirements specified in Appendix D of [PP_APP_V1.4].

2.1.9.2 Guidance Evaluation Activity

None.

2.1.9.3 Test Evaluation Activity

Performed in accordance with NIAP Policy Letter #5.

Section 6.1 of [ST] ("Cryptographic Support"), Table 4 ("Cryptographic Functions and CAVP Certificates") identifies the CAVP certifications verifying validation for deterministic random bit generation, as follows.

Algorithm	Tested Capabilities	Certificates
CTR_DRBG (AES) in accordance with NIST SP 800-90A.	Prediction Resistance: Yes, No Supports Reseed Capabilities: Mode: AES-128 Derivation Function Enabled: Yes Additional Input: 256 Entropy Input: 256 Nonce: 256 Personalization String Length: 256 Returned Bits: 512 Capabilities: Mode: AES-128 Derivation Function Enabled: No Additional Input: 256 Entropy Input: 256 Nonce: 256 Personalization String Length: 256 Returned Bits: 512 Capabilities: Mode: AES-192 Derivation Function Enabled: Yes Additional Input: 320 Entropy Input: 320 Nonce: 320 Personalization String Length: 320 Returned Bits: 512 Capabilities: Mode: AES-192 Derivation Function Enabled: No Additional Input: 320 Entropy Input: 320 Nonce: 320 Personalization String Length: 320 Returned Bits: 512 Capabilities: Mode: AES-256 Derivation Function Enabled: Yes Additional Input: 384 Entropy Input: 384 Nonce: 384 Personalization String Length: 384 Returned Bits: 512	CAVP #A2999 (Counter DRBG)

Algorithm	Tested Capabilities	Certificates
	Capabilities: Mode: AES-256 Derivation Function Enabled: No Additional Input: 384 Entropy Input: 384 Nonce: 384 Personalization String Length: 384 Returned Bits: 512	

2.1.10 FCS_STO_EXT.1 Storage of Credentials

2.1.10.1 TSS Evaluation Activity

The evaluator shall check the TSS to ensure that it lists all persistent credentials (secret keys, PKI private keys, or passwords) needed to meet the requirements in the ST. For each of these items, the evaluator shall confirm that the TSS lists for what purpose it is used, and how it is stored.

Section 6.1 of [ST] ("Cryptographic Support") states the TOE does not generate or store any credentials. All certificates and private keys are generated externally and are stored externally on the TOE platform.

2.1.10.2 Guidance Evaluation Activity

None.

2.1.10.3 Test Evaluation Activity

For all credentials for which the application implements functionality, the evaluator shall verify credentials are encrypted according to FCS_COP.1/SKC or conditioned according to FCS_CKM.1.1/AK and FCS_CKM.1/PBKDF. For all credentials for which the application invokes platform-provided functionality, the evaluator shall perform the following actions which vary per platform.

Platforms: Android… The evaluator shall verify that the application uses the Android KeyStore or the Android KeyChain to store certificates.

Platforms: Microsoft Windows... The evaluator shall verify that all certificates are stored in the Windows Certificate Store. The evaluator shall verify that other credentials, like passwords, are stored in the Windows Credential Manager or stored using the Data Protection API (DPAPI). For Windows Universal Applications, the evaluator shall verify that the application is using the ProtectData class and storing credentials in IsolatedStorage.

Platforms: Apple iOS… The evaluator shall verify that all credentials are stored within a Keychain.

Platforms: Linux… The evaluator shall verify that all credentials are stored using Linux keyrings.

Platforms: Apple macOS… The evaluator shall verify that all credentials are stored within Keychain.

The TOE does not store any credentials, so this Test activity is not applicable.

2.1.11 FCS_TLS_EXT.1 TLS Protocol (TLS Package)

2.1.11.1 TSS Evaluation Activity

None.

2.1.11.2 Guidance Evaluation Activity

The evaluator shall ensure that the selections indicated in the ST are consistent with selections in the dependent components.

The statement of FCS_TLS_EXT.1 in section 5.2.1 of [ST] ("Cryptographic Support (FCS)") selects only "TLS as a client". The evaluator confirmed the ST includes only SFRs specifying TLS client functionality (i.e., FCS_TLSC_EXT.* SFRs).

2.1.11.3 Test Evaluation Activity

None.

2.1.12 FCS_TLSC_EXT.1 TLS Client Protocol (TLS Package)

2.1.12.1 FCS_TLSC_EXT.1.1

2.1.12.1.1 TSS Evaluation Activity

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the cipher suites supported are specified. The evaluator shall check the TSS to ensure that the cipher suites specified include those listed for this component.

Section 6.1 of [ST] ("Cryptographic Support") lists the following cipher suites as being supported by the TOE:

- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384.

This list is consistent with the selections in FCS_TLSC_EXT.1.1.

2.1.12.1.2 Guidance Evaluation Activity

The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the product so that TLS conforms to the description in the TSS.

Section "Software Download and Installation", subsection "Installation" of [CCECG] provides instructions to place the TOE in its evaluated configuration. This includes enabling FIPS-CC mode, which restricts the TOE to using only the cipher suites claimed in the ST for TLS connections.

2.1.12.1.3 Test Evaluation Activity

The evaluator shall also perform the following tests:

Test 1: The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an EAP session. It is sufficient to observe the successful negotiation of a cipher suite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the cipher suite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

The evaluator verified that the TOE was successfully able to negotiate communication with the Server selecting each of the cipher suites specified by the ST.

Test 2: The goal of the following test is to verify that the TOE accepts only certificates with appropriate values in the extendedKeyUsage extension, and implicitly that the TOE correctly parses the extendedKeyUsage extension as part of X.509v3 server certificate validation.

The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension and verify that a connection is established. The evaluator shall repeat this test using a different, but otherwise valid and trusted, certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension and ensure that a connection is not established. Ideally, the two certificates should be similar in structure, the types of identifiers used, and the chain of trust.

The evaluator verified that the TOE accepted the TLS handshake that contained the proper EKU(ServerAuth). The evaluator repeated the connection, however this time the TOE rejected the TLS session because it did not contain the ServerAuth EKU.

Test 3: The evaluator shall send a server certificate in the TLS connection that does not match the server-selected cipher suite (for example, send a ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite or send a RSA certificate while using one of the ECDSA cipher suites.) The evaluator shall verify that the product disconnects after receiving the server's Certificate handshake message.

The evaluator verified that the TOE rejected connections when the server's certificate type did not match the cipher's authentication type.

Test 4: The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the client denies the connection.

The evaluator verified that the TOE rejected the TLS handshake once the server sent it's TLS_NULL_WITH_NULL_NULL cipher suite.

Test 5: The evaluator shall perform the following modifications to the traffic:

Test 5.1: Change the TLS version selected by the server in the Server Hello to an undefined TLS version (for example 1.5 represented by the two bytes 03 06) and verify that the client rejects the connection.

The evaluator verified that the TOE rejected the TLS version 1.5 (03 06).

Test 5.2: Change the TLS version selected by the server in the Server Hello to the most recent unsupported TLS version (for example 1.1 represented by the two bytes 03 02) and verify that the client rejects the connection.

The evaluator verified that the TOE rejected the most recent unsupported TLS version TLS 1.1.

Test 5.3: [conditional] If DHE or ECDHE cipher suites are supported, modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client does not complete the handshake and no application data flows.

The evaluator verified that the TOE did not complete ECDHE connections if the server's nonce was modified.

Test 5.4: Modify the server's selected cipher suite in the Server Hello handshake message to be a cipher suite not presented in the Client Hello handshake message. The evaluator shall verify that the client does not complete the handshake and no application data flows.

The evaluator verified that the TOE rejected connections where the server's selected cipher suite was not one offered by the TOE.

Test 5.5: [conditional] If DHE or ECDHE cipher suites are supported, modify the signature block in the server's Key Exchange handshake message, and verify that the client does not complete the handshake and no application data flows. This test does not apply to cipher suites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.

The evaluator verified that the TOE did not complete ECDHE connections when the signature block in the server's Key Exchange handshake message was modified.

Test 5.6: Modify a byte in the Server Finished handshake message, and verify that the client does not complete the handshake and no application data flows.

The evaluator verified that the TOE did not accept connections with a modified server finished message.

Test 5.7: Send a message consisting of random bytes from the server after the server has issued the Change Cipher Spec message and verify that the client does not complete the handshake and no application data flows. The message must still have a valid 5-byte record header in order to ensure the message will be parsed as TLS.

The evaluator verified that the TOE did not accept connections with random data for the server finished message.

2.1.12.2 FCS_TLSC_EXT.1.2

2.1.12.2.1 TSS Evaluation Activity

The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the application-configured reference identifier, including which types of reference identifiers are supported (e.g. Common Name, DNS Name, URI Name, Service Name, or other application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported. The evaluator shall ensure that this description identifies whether and the manner in which certificate pinning is supported or used by the product.

Section 6.1 of [ST] ("Cryptographic Support") states 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. The TOE does not support certificate pinning.

2.1.12.2.2 Guidance Evaluation Activity

The evaluator shall verify that the AGD guidance includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS.

Section "TOE Operation", subsection "Configuring the Portal and Gateway" of [CCECG] provides instructions for setting the reference identifier (i.e., the address of a portal, either IP address or hostname) to be used for the purposes of certificate validation in TLS.

2.1.12.2.3 Test Evaluation Activity

Modified in accordance with TD0499

The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. ***If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7***.

Test 1: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension.

The evaluator shall verify that the connection fails. Note that some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1.

The evaluator verified that the TOE rejected connections where the server presented a bad CN and no SAN.

Test 2: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type.

The evaluator verified that the TOE rejected connections where the server presented a matching CN to the reference identifier and a bad SAN.

Test 3: [conditional] If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.

The evaluator verified that the TOE accepted connections where the server presented a good CN and no SAN.

Test 4: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds.

The evaluator verified that the TOE accepted connections where the server presented a bad CN and a good SAN.

Test 5: The evaluator shall perform the following wildcard tests with each supported type of reference identifier. The support for wildcards is intended to be optional. If wildcards are supported, the first, second, and third tests below shall be executed. If wildcards are not supported, then the fourth test below shall be executed.

Test 5.1: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails.

The Evaluator verified that the TOE rejected connections from a server like foo.*.example.com.

Test 5.2: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.come) and verify that the connection fails.

The evaluator verified that the TOE accepted connections from a server like *.example.com, when the reference identifier was a singular ID and rejected connections from a server like *.example.com when the reference identifier was two id's or no id.

Test 5.3: [conditional]: If wildcards are supported, the evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.com) and verify that the connection fails.

The evaluator verified that the TOE rejected connections from a server like *.com, when the reference identifier was example.com or bar.example.com.

Test 5.4: [conditional]: If wildcards are not supported, the evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection fails.

Test 5.4 was not performed as wildcards are supported.

Test 6: [conditional] If URI or Service name reference identifiers are supported, the evaluator shall configure the DNS name and the service identifier. The evaluator shall present a server certificate containing the correct DNS name and service identifier in the URIName or SRVName fields of the SAN and verify that the connection succeeds. The evaluator shall repeat this test with the wrong service identifier (but correct DNS name) and verify that the connection fails.

The TOE does not claim support of URI or Server name reference identifiers and thus this test is not applicable.

Test 7: [conditional] If pinned certificates are supported the evaluator shall present a certificate that does not match the pinned certificate and verify that the connection fails.

The TOE does not claim support of pinned certificates and thus this test is not applicable.

2.1.12.3 FCS_TLSC_EXT.1.3

2.1.12.3.1 TSS Evaluation Activity

If the selection for authorizing override of invalid certificates is made, then the evaluator shall ensure that the TSS includes a description of how and when user or administrator authorization is obtained. The evaluator shall also ensure that the TSS describes any mechanism for storing such authorizations, such that future presentation of such otherwise-invalid certificates permits establishment of a trusted channel without user or administrator action.

The ST selects “with no exceptions” in FCS_TLSC_EXT.1.3, so this activity is not applicable to the TOE.

2.1.12.3.2 Guidance Evaluation Activity

None specified.

2.1.12.3.3 Test Evaluation Activity

The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted:

Modified in accordance with TD0513

Test 1a: *The evaluator shall demonstrate that a server using a certificate with a valid certification path successfully connects.*

Test 1b: *The evaluator shall modify the certificate chain used by the server in test 1a to be invalid and demonstrate that a server using a certificate without a valid certification path to a trust store element of the TOE results in an authentication failure.*

Test 1c *[conditional]: If the TOE trust store can be managed, the evaluator shall modify the trust store element used in Test 1a to be untrusted and demonstrate that a connection attempt from the same server used in Test 1a results in an authentication failure.*

The evaluator verified that the TOE only accepted certificates if the certificate trust store could validate the entire chain.

Test 2: The evaluator shall demonstrate that a server using a certificate which has been revoked results in an authentication failure.

The evaluator verified that the TOE rejected TLS session because the certificates was revoked by a revocation point.

Test 3: The evaluator shall demonstrate that a server using a certificate which has passed its expiration date results in an authentication failure.

The evaluator verified that the TOE rejected certificates that were past their expiration date/time.

Test 4: The evaluator shall demonstrate that a server using a certificate which does not have a valid identifier results in an authentication failure.

The evaluator verified that the TOE rejected certificates that did not contain valid identifiers for the server.

2.1.13 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication (TLS Package)

2.1.13.1 TSS Evaluation Activity

The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication. The evaluator shall also ensure that the TSS describes any factors beyond configuration that are necessary in order for the client to engage in mutual authentication using X.509v3 certificates.

Section 6.1 of [ST] ("Cryptographic Support") states client certificates can be set on the TOE to support TLS mutual authentication. The TOE does not require anything beyond correct configuration in order to engage in mutual authentication using X.509v3 certificates.

2.1.13.2 Guidance Evaluation Activity

The evaluator shall ensure that the AGD guidance includes any instructions necessary to configure the TOE to perform mutual authentication. The evaluator also shall verify that the AGD guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.

Section "PAN-OS Portal/Gateway Configuration" of [CCECG] provides instructions for configuring the GlobalProtect Gateway and Portal to enable the TOE to perform mutual authentication, including instructions to generate the client-side certificate used by the TOE and install the certificate on the TOE platform. The instructions cover all platforms included in the evaluation.

2.1.13.3 Test Evaluation Activity

The evaluator shall also perform the following tests:

Test 1: The evaluator shall establish a connection to a server that is not configured for mutual authentication (i.e. does not send Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE did not send Client's Certificate message (type 11) during handshake.

The evaluator verified that the TOE did not send it's client certificate upon receiving the server's response which lacked the Server Certificate Request (Type 13) message.

Test 2: The evaluator shall establish a connection to a server with a shared trusted root that is configured for mutual authentication (i.e. it sends Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message.

The evaluator verified that the TOE sent a non-empty client certificate upon receiving the server's response which contained the Server Certificate Request (Type 13) message.

2.1.14 FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension (TLS Package)

2.1.14.1 TSS Evaluation Activity

The evaluator shall verify that TSS describes the Supported Groups Extension.

Section 6.1 of [ST] states the TOE presents the Supported Groups Extension with the secp256r1 and secp384r1 curves.

2.1.14.2 Guidance Evaluation Activity

None.

2.1.14.3 Test Evaluation Activity

The evaluator shall also perform the following test:

Test 1: The evaluator shall configure a server to perform key exchange using each of the TOE's supported curves and/or groups. The evaluator shall verify that the TOE successfully connects to the server.

The evaluate confirmed that the TOE can connect to a TLS server which only supports secp256r1 and secp384r1 curves.

2.2 User Data Protection (FDP)

2.2.1 FDP_DAR_EXT.1 Encryption of Sensitive Application Data

2.2.1.1 TSS Evaluation Activity

The evaluator shall examine the TSS to ensure that it describes the sensitive data processed by the application. The evaluator shall then ensure that the following activities cover all of the sensitive data identified in the TSS.

If ***not store any sensitive data*** is selected, the evaluator shall inspect the TSS to ensure that it describes how sensitive data cannot be written to non-volatile memory. The evaluator shall also ensure that this is consistent with the filesystem test below.

The statement of FDP_DAR_EXT.1 in section 5.2.2 of [ST] ("User Data Protection (FDP)") selects "not store any sensitive data".

Section 6.2 of [ST] ("User Data Protection") states the TOE 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 a user is initiating connections to the Palo Alto Networks Gateway or Portal, they are required to enter their authentication data (which includes username and password defined on the Gateway/Portal) for the connection to succeed. These credentials are not stored or managed by the TOE.

2.2.1.2 Guidance Evaluation Activity

None.

2.2.1.3 Test Evaluation Activity

Evaluation activities (after the identification of the sensitive data) are to be performed on all sensitive data listed that are not covered by FCS_STO_EXT.1.

The TOE does not store sensitive data in non-volatile memory.

Modified in accordance with TD0756.

If "implement functionality to encrypt sensitive data as defined in the PP-Module for File Encryption" or "protect sensitive data in accordance with FCS_STO_EXT.1" is selected, the evaluator shall inventory the filesystem locations where the application may write data. The evaluator shall run the application and attempt to store sensitive data. The evaluator shall then inspect those areas of the filesystem to note where data was stored (if any), and determine whether it has been encrypted.

If ***leverage platform-provided functionality*** is selected, the evaluation activities will be performed as stated in the following requirements, which vary on a per-platform basis.

The ST does not select "leverage platform-provided functionality", so these test activities are not applicable.

2.2.2 FDP_DEC_EXT.1 Access to Platform Resources

2.2.2.1 FDP_DEC_EXT.1.1

2.2.2.1.1 TSS Evaluation Activity

None.

2.2.2.1.2 Guidance Evaluation Activity

The evaluator shall perform the platform-specific actions below and inspect user documentation to determine the application's access to hardware resources. The evaluator shall ensure that this is consistent with the selections indicated. The evaluator shall review documentation provided by the application developer and for each resource which it accesses, identify the justification as to why access is required.

The statement of FDP_DEC_EXT.1.1 in section 5.2.2 of [ST] ("User Data Protection (FDP)") specifies the TOE accesses network connectivity. Additionally, section 6.2 of [ST] ("User Data Protection") states the application accesses platform-provided network connectivity in order to communicate with the GlobalProtect Gateway/Portal. This is also stated in section "TOE Operation", subsection "Hardware Resources" of [CCECG], which provides justification why the TOE requires access to platform network connectivity hardware resources.

2.2.2.1.3 Test Evaluation Activity

Platforms: Android… The evaluator shall verify that each `uses-permission` entry in the `AndroidManifest.xml` file for access to a hardware resource is reflected in the selection.

The evaluator examined the permissions of the TOE and concluded that the TOE uses the proper permissions set.

Platforms: Microsoft Windows… For Windows Universal Applications the evaluator shall check the `WMAppManifest.xml` file for a list of required hardware capabilities. The evaluator shall verify that the user is made aware of the required hardware capabilities when the application is first installed. This includes permissions such as `ID_CAP_ISV_CAMERA`, `ID_CAP_LOCATION`, `ID_CAP_NETWORKING`, `ID_CAP_MICROPHONE`, `ID_CAP_PROXIMITY` and so on. A complete list of Windows App permissions can be found at:

http://msdn.microsoft.com/en-US/library/windows/apps/jj206936.aspx

For Windows Desktop Applications the evaluator shall identify in either the application software or its documentation the list of the required hardware resources.

Platforms: Apple iOS… The evaluator shall verify that either the application or the documentation provides a list of the hardware resources it accesses.

Platforms: Linux… The evaluator shall verify that either the application software or its documentation provides a list of the hardware resources it accesses.

Platforms: Apple macOS… The evaluator shall verify that either the application software or its documentation provides a list of the hardware resources it accesses.

The evaluator verified that the guidance documentations available make clear that The TOE makes use of the Network interfaces of the platform that the TOE is installed on.

2.2.2.2 FDP_DEC_EXT.1.2

2.2.2.2.1 TSS Evaluation Activity

None.

2.2.2.2.2 Guidance Evaluation Activity

The evaluator shall perform the platform-specific actions below and inspect user documentation to determine the application's access to sensitive information repositories. The evaluator shall ensure that this is consistent with the selections indicated. The evaluator shall review documentation provided by the application developer and for each sensitive information repository which it accesses, identify the justification as to why access is required.

The statement of FDP_DEC_EXT.1.1 in section 5.2.2 of [ST] (“User Data Protection (FDP)”) specifies the TOE does not access any sensitive platform information resources. In addition, section 6.2 of [ST] (“User Data Protection”) states the TOE does not require access to sensitive information repositories.

2.2.2.2.3 Test Evaluation Activity

Platforms: Android… The evaluator shall verify that each `uses-permission` entry in the `AndroidManifest.xml` file for access to a sensitive information repository is reflected in the selection.

The evaluator examined the permissions of the TOE and concluded that the TOE uses the proper permissions set.

Platforms: Microsoft Windows... For Windows Universal Applications the evaluator shall check the `WMAppManifest.xml` file for a list of required capabilities. The evaluator shall identify the required information repositories when the application is first installed. This includes permissions such as `ID_CAP_CONTACTS`, `ID_CAP_APPOINTMENTS`, `ID_CAP_MEDIALIB` and so on. A complete list of Windows App permissions can be found at:

http://msdn.microsoft.com/en-US/library/windows/apps/jj206936.aspx

For Windows Desktop Applications the evaluator shall identify in either the application software or its documentation the list of sensitive information repositories it accesses.

Platforms: Apple iOS… The evaluator shall verify that either the application software or its documentation provides a list of the sensitive information repositories it accesses.

Platforms: Linux… The evaluator shall verify that either the application software or its documentation provides a list of sensitive information repositories it accesses.

Platforms: Apple macOS… The evaluator shall verify that either the application software or its documentation provides a list of sensitive information repositories it accesses.

The tester verified that the guidance documentations available makes it clear that The TOE makes use of the Network interfaces of the platform that the TOE is installed on.

2.2.3 FDP_NET_EXT.1 Network Communications

2.2.3.1 TSS Evaluation Activity

None.

2.2.3.2 Guidance Evaluation Activity

None.

2.2.3.3 Test Evaluation Activity

The evaluator shall perform the following tests:

Test 1: The evaluator shall run the application. While the application is running, the evaluator shall sniff network traffic ignoring all non-application associated traffic and verify that any network communications witnessed are documented in the TSS or are user-initiated.

The evaluator sniffed the network traffic while the TOE was running in the background and noticed that no additional network connection was initiated, except for the platform-based services.

Test 2: The evaluator shall run the application. After the application initializes, the evaluator shall run network port scans to verify that any ports opened by the application have been captured in the ST for the third selection and its assignment. This includes connection-based protocols (e.g. TCP, DCCP) as well as connectionless protocols (e.g. UDP).

The evaluator ran a port scan while running the application and noticed that no ports were opened while the TOE was running, except for the platform-based services.

2.3 Identification and Authentication (FIA)

2.3.1 FIA_X509_EXT.1 X.509 Certificate Validation

2.3.1.1 FIA_X509_EXT.1.1

2.3.1.1.1 TSS Evaluation Activity

The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm.

Section 6.3 of [ST] (“Identification and Authentication”) states the TOE performs 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.

Section 6.3 of [ST] provides a description of the certificate path validation algorithm.

2.3.1.1.2 Guidance Evaluation Activity

None.

2.3.1.1.3 Test Evaluation Activity

The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created.

Test 1: The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function failing, for each of the following reasons, in turn:

- by establishing a certificate path in which one of the issuing certificates is not a CA certificate,
- by omitting the basicConstraints field in one of the issuing certificates,
- by setting the basicConstraints field in an issuing certificate to have CA=False,
- by omitting the CA signing bit of the key usage field in an issuing certificate, and
- by setting the path length field of a valid CA field to a value strictly less than the certificate path.

The evaluator shall then establish a valid certificate path consisting of valid CA certificates, and demonstrate that the function succeeds. The evaluator shall then remove trust in one of the CA certificates, and show that the function fails.

The evaluator verified that the TOE will not validate a certificate without a valid certification path and will validate a certificate that has a valid path.

Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.

The evaluator verified that the TOE rejects expired certificate.

Test 3: The evaluator shall test that the TOE can properly handle revoked certificates-–conditional on whether CRL, OCSP, OCSP Stapling or OCSP Multi-stapling is selected; if multiple methods are selected, then the following tests shall be performed for each method:

- The evaluator shall test revocation of the node certificate.
- The evaluator shall also test revocation of an intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA), if intermediate CA certificates are supported. If OCSP Stapling per RFC 6066 is the only supported revocation method, this test is omitted.
- The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails.

The TOE was shown to reject CRL and OCSP certificates that were revoked both at the end entity level and the intermediate CA level.

Modified in accordance with TD0669.

Test 4: If any OCSP option is selected, the evaluator shall ensure the TSF has no other source of revocation information available and configure the OCSP server or use a man-in-the-middle tool to present an OCSP response signed by a certificate that does not have the OCSP signing purpose and which is the only source of revocation status information advertised by the CA issuing the certificate being validated. The evaluator shall verify that validation of the OCSP response fails and that the TOE treats the certificate being checked as invalid and rejects the connection. If CRL is selected, the evaluator shall likewise configure the CA to be the only source of revocation status information, and sign a CRL with a certificate that does not have the cRLsign key usage bit set~~, and~~. The evaluator shall verify that validation of the CRL fails and that the TOE treats the certificate being checked as invalid and rejects the connection.

Note: The intent of this test is to ensure a TSF does not trust invalid revocation status information. A TSF receiving invalid revocation status information from the only advertised certificate status provider should treat the certificate whose status is being checked as invalid. This should generally be treated differently from the case where the TSF is not able to establish a connection to check revocation status information, but it is acceptable that the TSF ignore any invalid information and attempt to find another source of revocation status (another advertised provider, a locally configured provider, or cached information) and treat this situation as not having a connection to a valid certificate status provider.

The TOE would not validate a certificate whose OCSP responder did not have the OCSP signing purpose in its certificate. Additionally, if a CRL was signed by a CA whose certificate lacked the CRLsign key usage bit the TOE would not validate that CRL.

Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)

The TOE terminated the session after receiving a certificate with the first eight bytes modified.

Test 6: The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

The TOE terminated the session after receiving a certificate with the last eight bytes modified.

Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

The TOE terminated the session after receiving a certificate modified public key from the TLS server.

Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/Sig). The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

The TOE accepted the session after receiving an EC certificate chain from the TLS server.

Test 9: (Conditional on support for EC certificates as indicated in FCS_COP.1/Sig). The evaluator shall replace the intermediate certificate in the certificate chain for Test 8a with a modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 8a, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

The TOE terminated the session after receiving an EC certificate chain from the TLS server that had an explicit EC parameter in one of the CA that was provided by the server.

2.3.1.2 FIA_X509_EXT.1.2

2.3.1.2.1 TSS Evaluation Activity

None.

2.3.1.2.2 Guidance Evaluation Activity

None.

2.3.1.2.3 Test Evaluation Activity

The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created.

Test 1: The evaluator shall ensure that the certificate of at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator shall confirm that validation of the certificate path fails (i) as part of the validation of the peer certificate belonging to this chain; and/or (ii) when attempting to add the CA certificate without the basicConstraints extension to the TOE's trust store.

The evaluator verified that the TOE will not accept a certificate issued by a CA whose certificate does not have the basicConstraints extension.

Test 2: The evaluator shall ensure that the certificate of at least one of the CAs in the chain has the CA flag in the basicConstraints extension not set (or set to FALSE). The evaluator shall confirm that validation of the certificate path fails (i) as part of the validation of the peer certificate belonging to this chain; and/or (ii) when attempting to add the CA certificate with the CA flag not set (or set to FALSE) in the basicConstraints extension to the TOE's trust store.

The evaluator verified that the TOE will not accept a certificate issued by a CA whose certificate has a basicConstraints set to false.

2.3.2 FIA_X509_EXT.2 X.509 Certificate Authentication

2.3.2.1 TSS Evaluation Activity

The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates.

Section 6.3 of [ST] ("Identification and Authentication") states the TOE performs certificate path validation on the certificate chain 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.

The topic "Create necessary certificates" in section "PAN-OS Portal/Gateway Configuration", subsection "Enable configuration settings on the Palo Alto Networks Firewall" provides instructions for configuring the operating environment so the TOE can use the certificates. It covers generating leaf and CA certificates for the PAN-OS device acting as the GlobalProtect gateway, generating certificates for GlobalProtect

clients, and installing all certificates on the TOE platform. The guidance addresses all claimed platforms in the evaluated configuration.

The evaluator shall examine the TSS to confirm that it describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described.

Section 6.3 of [ST] states, in the event a connection cannot be established during the validity check of a certificate, the TOE provides a warning message to the administrator that revocation status could not be determined, along with the option to proceed with the connection. The TOE supports the single trusted channel between the TOE and the Palo Alto Networks GlobalProtect Gateway and Portal.

If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed.

The TOE does not provide the ability for the administrator to specify the default action in the event the revocation status of a certificate cannot be determined.

2.3.2.2 Guidance Evaluation Activity

None.

2.3.2.3 Test Evaluation Activity

The evaluator shall perform the following test for each trusted channel:

Test 1: The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the operational guidance to determine that all supported administrator-configurable options behave in their documented manner.

Test 2: The evaluator shall demonstrate that an invalid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity cannot be accepted.

The evaluator verified that certificates requiring validation checking were able to be checked against a non-TOE IT entity, and that when the IT entity was unavailable, the TOE terminated the session properly.

2.4 Security Management (FMT)

2.4.1 FMT_CFG_EXT.1 Secure by Default Configuration

2.4.1.1 FMT_CFG_EXT.1.1

2.4.1.1.1 TSS Evaluation Activity

The evaluator shall check the TSS to determine if the application requires any type of credentials and if the application installs with default credentials.

Section 6.4 of [ST] (“Security Management”) states the TOE does not require any credentials and does not install with any default credentials.

2.4.1.1.2 Guidance Evaluation Activity

None.

2.4.1.1.3 Test Evaluation Activity

If the application uses any default credentials the evaluator shall run the following tests.

Test 1: The evaluator shall install and run the application without generating or loading new credentials and verify that only the minimal application functionality required to set new credentials is available.

Test 2: The evaluator shall attempt to clear all credentials and verify that only the minimal application functionality required to set new credentials is available.

Test 3: The evaluator shall run the application, establish new credentials and verify that the original default credentials no longer provide access to the application.

As stated in section 6.4 of [ST], the TOE does not use any default credentials. Therefore, this test activity is not applicable to the TOE.

2.4.1.2 FMT_CFG_EXT.1.2

2.4.1.2.1 TSS Evaluation Activity

None.

2.4.1.2.2 Guidance Evaluation Activity

None.

2.4.1.2.3 Test Evaluation Activity

The evaluator shall install and run the application. The evaluator shall inspect the filesystem of the platform (to the extent possible) for any files created by the application and ensure that their permissions are adequate to protect them. The method of doing so varies per platform.

Platforms: Android… The evaluator shall run the command `find -L . -perm /002` inside the application's data directories to ensure that all files are not world-writable. The command should not print any files.

Platforms: Microsoft Windows... The evaluator shall run the SysInternals tools, Process Monitor and Access Check (or tools of equivalent capability, like icacls.exe) for Classic Desktop applications to verify that files written to disk during an application's installation have the correct file permissions, such that a standard user cannot modify the application or its data files. For Windows Universal Applications the evaluator shall consider the requirement met because of the AppContainer sandbox.

Platforms: Apple iOS… The evaluator shall determine whether the application leverages the appropriate Data Protection Class for each data file stored locally.

Platforms: Linux… The evaluator shall run the command `find -L . -perm /002` inside the application's data directories to ensure that all files are not world-writable. The command should not print any files.

Platforms: Apple macOS… The evaluator shall run the command `find . -perm +002` inside the application's data directories to ensure that all files are not world-writable. The command should not print any files.

The evaluator inspected the file system of the TOE and ensured that it had the proper permissions set.

2.4.2 FMT_MEC_EXT.1 Supported Configuration Mechanism

2.4.2.1 TSS Evaluation Activity

The evaluator shall review the TSS to identify the application's configuration options (e.g. settings) and determine whether these are stored and set using the mechanisms supported by the platform or implemented by the application in accordance with the PP-Module for File Encryption. At a minimum the TSS shall list settings related to any SFRs and any settings that are mandated in the operational guidance in response to an SFR.

The evaluator reviewed the TSS and determined the TOE's configuration options consist of the Gateway and Portal addresses. Section 6.4 of [ST] ("Security Management") states the TOE stores configuration data using mechanisms recommended by the OS and describes the process to enable FIPS-CC mode on each supported platform. It also states the TOE provides several management functions, including setting gateway and portal addresses.

Conditional: If "***implement functionality to encrypt and store configuration options as defined by FDP_PRT_EXT.1 in the PP-Module for File Encryption***" is selected, the evaluator shall ensure that the TSS identifies those options, as well as indicates where the encrypted representation of these options is stored.

The ST does not make this selection, so this activity is not applicable.

2.4.2.2 Guidance Evaluation Activity

None.

2.4.2.3 Test Evaluation Activity

Modified in accordance with TD0624.

If "***invoke the mechanisms recommended by the platform vendor for storing and setting configuration options***" is chosen, the method of testing varies per platform as follows:

Platforms: Android… The evaluator shall run the application and make security-related changes to its configuration. The evaluator shall check that at least one ~~XML~~ file exists at location `/data/data/package/shared_prefs/`(for SharedPreferences) and/or `/data/data/package/files/datastore` (for DataStore) where the package is the Java package of the application. For SharedPreferences the evaluator shall examine the XML file to make sure it reflects the changes made to the configuration to verify that the application used SharedPreferences and/or PreferenceActivity to store the configuration data. For DataStore the evaluator shall use a protocol buffer analyzer to examine the file to make sure it reflects the changes made to the configuration to verify that the application used DataStore to store the configuration data.

Platforms: Microsoft Windows... The evaluator shall determine and verify that Windows Universal Applications use either the Windows.Storage namespace, Windows.UI.ApplicationSettings namespace, or the IsolatedStorageSettings namespace for storing application specific settings. For .NET applications, the evaluator shall determine and verify that the application uses one of the locations listed in https://docs.microsoft.com/en-us/dotnet/framework/configure-apps/ for storing application specific settings. For Classic Desktop applications, the evaluator shall run the application while monitoring it with the SysInternals tool ProcMon and make changes to its configuration. The evaluator shall verify that ProcMon logs show corresponding changes to the Windows Registry or `C:\ProgramData\` directory.

Platforms: Apple iOS… The evaluator shall verify that the app uses the `user defaults system` or `key-value store` for storing all settings.

Platforms: Linux… The evaluator shall run the application while monitoring it with the utility `strace`. The evaluator shall make security-related changes to its configuration. The evaluator shall verify that `strace` logs corresponding changes to configuration files that reside in `/etc` (for system-specific configuration), in the user's home directory (for user-specific configuration), or `/var/lib/` (for configurations controlled by UI and not intended to be directly modified by an administrator).

Platforms: Apple macOS… The evaluator shall verify that the application stores and retrieves settings using the `NSUserDefaults` class.

If "***implement functionality to encrypt and store configuration options as defined by FDP_PRT_EXT.1 in the PP-Module for File Encryption***" is selected, for all configuration options listed in the TSS as being stored and protected using encryption, the evaluator shall examine the contents of the configuration option storage (identified in the TSS) to determine that the options have been encrypted.

The evaluator verified that the TOE stored the configuration values in a secure manner as described by this requirement for each platform.

2.4.3 FMT_SMF.1 Specification of Management Functions

2.4.3.1 TSS Evaluation Activity

None.

2.4.3.2 Guidance Evaluation Activity

The evaluator shall verify that every management function mandated by the PP is described in the operational guidance and that the description contains the information required to perform the management duties associated with the management function.

The statement of FMT_SMF.1 in section 5.2.4 of [ST] (“Security Management (FMT)”) specifies the following security management functions:

- Enable/disable transmission of any information describing the system’s hardware, software, or configuration
- Set gateway and portal addresses
- Collect troubleshooting logs
- Check for updates
- Query the current version of the TOE.

Section “TOE Operation” of [CCECG] describes each of these security management functions, as follows:

- Subsection “Sending System Information to Portal” describes how the administrator enables transmission of system information to the GlobalProtect portal
- Subsection “Configuring the Portal and Gateway” describes how the administrator configures the IP address or hostname of a portal to connect to, and how to select which configured gateways and portals to connect to
- Subsection “Collecting Logs” describes how the administrator can collect application logs for troubleshooting purposes
- Subsection “Checking for Updates” describes how the administrator can check if updates to the TOE are available to download and install

- Subsection “Viewing the Current Version” describes how the administrator is able to query the current version of the TOE.

2.4.3.3 Test Evaluation Activity

The evaluator shall test the application's ability to provide the management functions by configuring the application and testing each option selected from above. The evaluator is expected to test these functions in all the ways in which the ST and guidance documentation state the configuration can be managed.

For each of the management functions defined by the ST, the evaluator verified that the TOE was able to perform the management actions as described in the AGD.

2.5 Privacy (FPR)

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

2.5.1.1 TSS Evaluation Activity

The evaluator shall inspect the TSS documentation to identify functionality in the application where PII can be transmitted.

Section 6.5 of [ST] (“Privacy”) states the TOE 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.

2.5.1.2 Guidance Evaluation Activity

None.

2.5.1.3 Test Evaluation Activity

If ***require user approval before executing*** is selected, the evaluator shall run the application and exercise the functionality responsible for transmitting PII and verify that user approval is required before transmission of the PII.

The ST does not select “require user approval before executing”. As such, this test activity is not applicable to the TOE.

2.6 Protection of the TSF (FPT)

2.6.1 FPT_AEX_EXT.1 Anti-Exploitation Capabilities

2.6.1.1 FPT_AEX_EXT.1.1

2.6.1.1.1 TSS Evaluation Activity

The evaluator shall ensure that the TSS describes the compiler flags used to enable ASLR when the application is compiled.

Section 6.6 of [ST] (“Protection of the TSF”) states the Windows version of the TOE is compiled with the `/DYNAMICBASE` flag and the macOS, iOS, Android, and Linux versions of the TOE are compiled with the `-pie` flag.

2.6.1.1.2 Guidance Evaluation Activity

None.

2.6.1.1.3 Test Evaluation Activity

The evaluator shall perform either a static or dynamic analysis to determine that no memory mappings are placed at an explicit and consistent address. The method of doing so varies per platform. For those platforms requiring the same application running on two different systems, the evaluator may alternatively use the same device. After collecting the first instance of mappings, the evaluator must uninstall the application, reboot the device, and reinstall the application to collect the second instance of mappings.

Platforms: Android… The evaluator shall run the same application on two different Android systems. Both devices do not need to be evaluated, as the second device is acting only as a tool. Connect via ADB and inspect `/proc/PID/maps`. Ensure the two different instances share no memory mappings made by the application at the same location.

Platforms: Microsoft Windows... The evaluator shall run the same application on two different Windows systems and run a tool that will list all memory mapped addresses for the application. The evaluator shall then verify the two different instances share no mapping locations. The Microsoft SysInternals tool, VMMap, could be used to view memory addresses of a running application. The evaluator shall use a tool such as Microsoft's BinScope Binary Analyzer to confirm that the application has ASLR enabled.

Platforms: Apple iOS... The evaluator shall perform a static analysis to search for any `mmap` calls (or API calls that call `mmap`), and ensure that no arguments are provided that request a mapping at a fixed address.

Platforms: Linux… The evaluator shall run the same application on two different Linux systems. The evaluator shall then compare their memory maps using `pmap -x PID` to ensure the two different instances share no mapping locations.

Platforms: Apple macOS… The evaluator shall run the same application on two different Mac systems. The evaluator shall then compare their memory maps using `vmap PID` to ensure the two different instances share no mapping locations.

The evaluator verified that the TOE did not map any memory address. For some platforms, the only mapped memory that was detect, turned out to be made by the platform's default service.

2.6.1.2 FPT_AEX_EXT.1.2

2.6.1.2.1 TSS Evaluation Activity

None.

2.6.1.2.2 Guidance Evaluation Activity

None.

2.6.1.2.3 Test Evaluation Activity

The evaluator shall verify that no memory mapping requests are made with write and execute permissions. The method of doing so varies per platform.

Platforms: Android… The evaluator shall perform static analysis on the application to verify that

- `mmap` is never be invoked with both the `PROT_WRITE` and `PROT_EXEC` permissions, and

- `mprotect` is never invoked.

Platforms: Microsoft Windows... The evaluator shall use a tool such as Microsoft's BinScope Binary Analyzer to confirm that the application passes the NXCheck. The evaluator may also ensure that the `/NXCOMPAT` flag was used during compilation to verify that DEP protections are enabled for the application.

Platforms: Apple iOS... The evaluator shall perform static analysis on the application to verify that `mprotect` is never invoked with the `PROT_EXEC` permission.

Platforms: Linux… The evaluator shall perform static analysis on the application to verify that both

- `mmap` is never be invoked with both the `PROT_WRITE` and `PROT_EXEC` permissions, and
- `mprotect` is never invoked with the `PROT_EXEC` permission.

Platforms: Apple macOS... The evaluator shall perform static analysis on the application to verify that `mprotect` is never invoked with the `PROT_EXEC` permission.

The evaluator verified that the TOE's source code is free of any of the constraint that specified by the SFR.

2.6.1.3 FPT_AEX_EXT.1.3

2.6.1.3.1 TSS Evaluation Activity

None.

2.6.1.3.2 Guidance Evaluation Activity

None.

2.6.1.3.3 Test Evaluation Activity

The evaluator shall configure the platform in the ascribed manner and carry out one of the prescribed tests:

Platforms: Android… Applications running on Android cannot disable Android security features, therefore this requirement is met and no evaluation activity is required.

Platforms: Microsoft Windows... If the OS platform supports Windows Defender Exploit Guard (Windows 10 version 1709 or later), then the evaluator shall ensure that the application can run successfully with Windows Defender Exploit Guard Exploit Protection configured with the following minimum mitigations enabled; Control Flow Guard (CFG), Randomize memory allocations (Bottom-Up ASLR), Export address filtering (EAF), Import address filtering (IAF), and Data Execution Prevention (DEP). The following link describes how to enable Exploit Protection,

https://docs.microsoft.com/en-us/windows/security/threat-protection/windows-defender-exploit-guard/customize-exploit-protection.

If the OS platform supports the Enhanced Mitigation Experience Toolkit (EMET) which can be installed on Windows 10 version 1703 and earlier, then the evaluator shall ensure that the application can run successfully with EMET configured with the following minimum mitigations enabled; Memory Protection Check, Randomize memory allocations (Bottom-Up ASLR), Export address filtering (EAF), and Data Execution Prevention (DEP).

Platforms: Apple iOS… Applications running on iOS cannot disable security features, therefore this requirement is met and no evaluation activity is required.

Platforms: Linux… The evaluator shall ensure that the application can successfully run on a system with either SELinux or AppArmor enabled and in enforce mode.

Platforms: Apple macOS… The evaluator shall ensure that the application can successfully run on macOS without disabling any security features.

For the Windows and Linux platform, the evaluator verified that the TOE was able to successfully run on the platform with the described platform protections enabled.

There are no platform security features on MacOS that can be disabled. No changes to the OS were required to install and run the TOE. This requirement is met.

Applications that run on the iOS and Android cannot operate without the security functionality that is enabled by the platform, therefor the requirement is met.

2.6.1.4 FPT_AEX_EXT.1.4

2.6.1.4.1 TSS Evaluation Activity

None.

2.6.1.4.2 Guidance Evaluation Activity

None.

2.6.1.4.3 Test Evaluation Activity

The evaluator shall run the application and determine where it writes its files. For files where the user does not choose the destination, the evaluator shall check whether the destination directory contains executable files. This varies per platform:

Platforms: Android… The evaluator shall run the program, mimicking normal usage, and note where all user-modifiable files are written. The evaluator shall ensure that there are no executable files stored under `/data/data/package/` where `package` is the Java package of the application.

Platforms: Microsoft Windows... For Windows Universal Applications the evaluator shall consider the requirement met because the platform forces applications to write all data within the application working directory (sandbox). For Windows Desktop Applications the evaluator shall run the program, mimicking normal usage, and note where all user-modifiable files are written. The evaluator shall ensure that there are no executable files stored in the same directories to which the application wrote user-modifiable files.

Platforms: Apple iOS… The evaluator shall consider the requirement met because the platform forces applications to write all data within the application working directory (sandbox).

Platforms: Linux… The evaluator shall run the program, mimicking normal usage, and note where all user-modifiable files are written. The evaluator shall ensure that there are no executable files stored in the same directories to which the application wrote user-modifiable files.

Platforms: Apple macOS… The evaluator shall run the program, mimicking normal usage, and note where all user-modifiable files are written. The evaluator shall ensure that there are no executable files stored in the same directories to which the application wrote user-modifiable files.

The evaluator verified that the TOE stored files created during operations in the manner prescribed for each platform.

2.6.1.5 FPT_AEX_EXT.1.5

2.6.1.5.1 TSS Evaluation Activity

None.

2.6.1.5.2 Guidance Evaluation Activity

None.

2.6.1.5.3 Test Evaluation Activity

The evaluator will inspect every native executable included in the TOE to ensure that stack-based buffer overflow protection is present.

Platforms: Microsoft Windows… Applications that run as Managed Code in the .NET Framework do not require these stack protections. Applications developed in Object Pascal using the Delphi IDE compiled with RangeChecking enabled comply with this element. For other code, the evaluator shall review the TSS and verify that the `/GS` flag was used during compilation. The evaluator shall run a tool like, BinScope, that can verify the correct usage of `/GS`.

For PE , the evaluator will disassemble each and ensure the following sequence appears:

```
mov rcx, QWORD PTR [rsp+(...)]
xor rcx, (...)
call (...)
```

For ELF executables, the evaluator will ensure that each contains references to the symbol `__stack_chk_fail`.

Tools such as Canary Detector may help automate these activities.

The evaluator verified that the TOE's executables on the windows system passed the /GS checks.

2.6.2 FPT_API_EXT.1 Use of Supported Services and APIs

2.6.2.1 TSS Evaluation Activity

The evaluator shall verify that the TSS lists the platform APIs used in the application.

The vendor provided the list of platform APIs used by each platform version of the TOE in a separate proprietary appendix to the ST.

2.6.2.2 Guidance Evaluation Activity

None.

2.6.2.3 Test Evaluation Activity

The evaluator shall then compare the list with the supported APIs (available through e.g. developer accounts, platform developer groups) and ensure that all APIs listed in the TSS are supported.

The evaluator searched platform documentation for the API's provided and found API references for each API provided.

2.6.3 FPT_LIB_EXT.1 Use of Third Party Libraries

2.6.3.1 TSS Evaluation Activity

None.

2.6.3.2 Guidance Evaluation Activity

None.

2.6.3.3 Test Evaluation Activity

The evaluator shall install the application and survey its installation directory for dynamic libraries. The evaluator shall verify that libraries found to be packaged with or employed by the application are limited to those in the assignment.

The evaluator performed a recursive lookup of the TOE's directory and surveyed the dynamic libraries that were present in the subdirectories and verified that the items that were listed were limited to the libraries claimed in the ST.

2.6.4 FPT_IDV_EXT.1 Software Identification and Versions

2.6.4.1 TSS Evaluation Activity

If **"other version information"** is selected the evaluator shall verify that the TSS contains an explanation of the versioning methodology.

The statement of FPT_IDV_EXT.1 in section 5.2.6 of [ST] ("Protection of the TSF (FPT)") selects "other version information" and completes the assignment with "GlobalProtect software version".

Section 6.6 of [ST] ("Protection of the TSF") states 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. 6.0.1, 6.0.2) are released on a regular cadence to fix issues identified with the major release.

2.6.4.2 Guidance Evaluation Activity

None.

2.6.4.3 Test Evaluation Activity

The evaluator shall install the application, then check for the existence of version information. If ***SWID tags*** is selected the evaluator shall check for a .swidtag file. The evaluator shall open the file and verify that is contains at least a SoftwareIdentity element and an Entity element.

The evaluator verified that the TOE is able to provide version information of the TOE.

2.6.5 FPT_TUD_EXT.1 Integrity for Installation and Update

2.6.5.1 FPT_TUD_EXT.1.1

2.6.5.1.1 TSS Evaluation Activity

None.

2.6.5.1.2 Guidance Evaluation Activity

The evaluator shall check to ensure the guidance includes a description of how updates are performed.

Section “TOE Operation”, subsection “Checking for Updates” of [CCECG] describes how to check for updates to the TOE and how to initiate the update procedure.

2.6.5.1.3 Test Evaluation Activity

The evaluator shall check for an update using procedures described in either the application documentation or the platform documentation and verify that the application does not issue an error. If it is updated or if it reports that no update is available this requirement is considered to be met.

The evaluator was able to check to see if there was an update for the application as described in the AGD.

2.6.5.2 FPT_TUD_EXT.1.2

2.6.5.2.1 TSS Evaluation Activity

None.

2.6.5.2.2 Guidance Evaluation Activity

The evaluator shall verify guidance includes a description of how to query the current version of the application.

Section “TOE Operation”, subsection “Viewing the Current Version” of [CCECG] describes how to query the current version of the TOE, by opening the “Settings” dialog and selecting “About”.

2.6.5.2.3 Test Evaluation Activity

The evaluator shall query the application for the current version of the software according to the operational user guidance. The evaluator shall then verify that the current version matches that of the documented and installed version.

The evaluator verified that the TOE is able to provide version information of the TOE.

2.6.5.3 FPT_TUD_EXT.1.3

2.6.5.3.1 TSS Evaluation Activity

None.

2.6.5.3.2 Guidance Evaluation Activity

None.

2.6.5.3.3 Test Evaluation Activity

The evaluator shall verify that the application's executable files are not changed by the application.

Platforms: Apple iOS… The evaluator shall consider the requirement met because the platform forces applications to write all data within the application working directory (sandbox).

For all other platforms, the evaluator shall perform the following test:

Test 1: The evaluator shall install the application and then locate all of its executable files. The evaluator shall then, for each file, save off either a hash of the file or a copy of the file itself. The evaluator shall then run the application and exercise all features of the application as described in the ST. The evaluator shall then compare each executable file with the either the saved hash or the saved copy of the files. The evaluator shall verify that these are identical.

The evaluator used a hashing technique to verify that the TOE does not modify the executable as it's executed.

2.6.5.4 FPT_TUD_EXT.1.4

2.6.5.4.1 TSS Evaluation Activity

The evaluator shall verify that the TSS identifies how updates to the application are signed by an authorized source. The definition of an authorized source must be contained in the TSS. The evaluator shall also ensure that the TSS (or the operational guidance) describes how candidate updates are obtained.

Section 6.6 of [ST] ("Protection of the TSF") states Palo Alto Networks digitally signs all new versions of the TOE using RSA 2048 with SHA-256.

Section "TOE Operation", subsection "Checking for Updates" of [CCECG] describes how to check for updates to the TOE and how to initiate the update procedure.

2.6.5.4.2 Guidance Evaluation Activity

None.

2.6.5.4.3 Test Evaluation Activity

None.

2.6.5.5 FPT_TUD_EXT.1.5

2.6.5.5.1 TSS Evaluation Activity

The evaluator shall verify that the TSS identifies how the application is distributed. If "***with the platform***" is selected the evaluated shall perform a clean installation or factory reset to confirm that TOE software is included as part of the platform OS. If "***as an additional package***" is selected the evaluator shall perform the tests in FPT_TUD_EXT.2.

Section 6.6 of [ST] ("Protection of the TSF") states the TOE is distributed using the following platform-specific package formats:

- Windows 10--<filename>.msi
- macOS--<filename>.pkg
- Android--<filename>.apk

- Linux--<filename>.tgz
- iOS—downloaded from iTunes Store.

2.6.5.5.2 Guidance Evaluation Activity

None.

2.6.5.5.3 Test Evaluation Activity

None.

2.6.6 FPT_TUD_EXT.2 Integrity for Installation and Update

2.6.6.1 FPT_TUD_EXT.2.1

2.6.6.1.1 TSS Evaluation Activity

None.

2.6.6.1.2 Guidance Evaluation Activity

None.

2.6.6.1.3 Test Evaluation Activity

Modified in accordance with TD0628.

If a container image is claimed the evaluator shall verify that application updates are distributed as container images. If the format of the platform-supported package manager is claimed, the evaluator shall verify that application updates are distributed in the format supported by the platform. This varies per platform:

Platforms: Android… The evaluator shall ensure that the application is packaged in the Android application package (APK) format.

Platforms: Microsoft Windows... The evaluator shall ensure that the application is packaged in the standard Windows Installer (.MSI) format, the Windows Application Software (.EXE) format signed using the Microsoft Authenticode process, or the Windows Universal Application package (.APPX) format. See https://msdn.microsoft.com/enus/library/ms537364(v=vs.85).aspx for details regarding Authenticode signing.

Platforms: Apple iOS… The evaluator shall ensure that the application is packaged in the IPA format.

Platforms: Linux... The evaluator shall ensure that the application is packaged in the format of the package management infrastructure of the chosen distribution. For example, applications running on Red Hat and Red Hat derivatives shall be packaged in RPM format. Applications running on Debian and Debian derivatives shall be packaged in DEB format.

Platforms: Apple macOS… The evaluator shall ensure that the application is packaged in the DMG format, the PKG format, or the MPKG format

The evaluator verified that the TOE was packaged in the format described for each platform.

2.6.6.2 FPT_TUD_EXT.2.2

2.6.6.2.1 TSS Evaluation Activity

None.

2.6.6.2.2 Guidance Evaluation Activity

None.

2.6.6.2.3 Test Evaluation Activity

Modified in accordance with TD0664.

Platforms: Android… The evaluator shall consider the requirement met because the platform forces applications to write all data within the application working directory (sandbox).

~~**Platforms: Microsoft Windows…** The evaluator shall install the application and then locate all of its executable files. The evaluator shall then, for each file, save off either a hash of the file or a copy of the file itself. The evaluator shall then run the application and exercise all features of the application as described in the ST. The evaluator shall then compare each executable file with the either the saved hash or the saved copy of the files. The evaluator shall verify that these are identical.~~

Platforms: Apple iOS… The evaluator shall consider the requirement met because the platform forces applications to write all data within the application working directory (sandbox).

~~**Platforms: Linux…** The evaluator shall install the application and then locate all of its executable files. The evaluator shall then, for each file, save off either a hash of the file or a copy of the file itself. The evaluator shall then run the application and exercise all features of the application as described in the ST. The evaluator shall then compare each executable file with the either the saved hash or the saved copy of the files. The evaluator shall verify that these are identical.~~

~~**Platforms: Apple MacOS…**~~

~~The evaluator shall install the application and then locate all of its executable files. The evaluator shall then, for each file, save off either a hash of the file or a copy of the file itself. The evaluator shall then run the application and exercise all features of the application as described in the ST. The evaluator shall then compare each executable file with the either the saved hash or the saved copy of the files. The evaluator shall verify that these are identical.~~

All Other Platforms… The evaluator shall record the path of every file on the entire filesystem prior to installation of the application, and then install and run the application. Afterwards, the evaluator shall then uninstall the application, and compare the resulting filesystem to the initial record to verify that no files, other than configuration, output, and audit/log files, have been added to the filesystem.

The evaluator verified that the TOE was able to successfully install and uninstall itself, while only keeping miscellaneous config files and logs of the TOE.

2.6.6.3 FPT_TUD_EXT.2.3

2.6.6.3.1 TSS Evaluation Activity

The evaluator shall verify that the TSS identifies how the application installation package is signed by an authorized source. The definition of an authorized source must be contained in the TSS.

Section 6.6 of [ST] (“Protection of the TSF”) states Palo Alto Networks digitally signs all new versions of the TOE using RSA 2048 with SHA-256.

2.6.6.3.2 Guidance Evaluation Activity

None.

2.6.6.3.3 Test Evaluation Activity

None.

2.7 Trusted Path/Channels (FTP)

2.7.1 FTP_DIT_EXT.1 Protection of Data in Transit

2.7.1.1 TSS Evaluation Activity

For platform-provided functionality, the evaluator shall verify the TSS contains the calls to the platform that TOE is leveraging to invoke the functionality.

The statement of FTP_DIT_EXT.1 in section 5.2.7 of [ST] (“Trusted Path/Channel (FTP)”) specifies the TOE encrypts all transmitted data between itself and other trusted IT products and does not rely on platform-provided functionality. Therefore, this activity is not applicable.

2.7.1.2 Guidance Evaluation Activity

None.

2.7.1.3 Test Evaluation Activity

The evaluator shall perform the following tests:

Test 1: The evaluator shall exercise the application (attempting to transmit data; for example by connecting to remote systems or websites) while capturing packets from the application. The evaluator shall verify from the packet capture that the traffic is encrypted with HTTPS, TLS, DTLS, SSH, or IPsec in accordance with the selection in the ST.

Test 2: The evaluator shall exercise the application (attempting to transmit data; for example by connecting to remote systems or websites) while capturing packets from the application. The evaluator shall review the packet capture and verify that no sensitive data is transmitted in the clear.

Test 3: The evaluator shall inspect the TSS to determine if user credentials are transmitted. If credentials are transmitted the evaluator shall set the credential to a known value. The evaluator shall capture packets from the application while causing credentials to be transmitted as described in the TSS. The evaluator shall perform a string search of the captured network packets and verify that the plaintext credential previously set by the evaluator is not found.

The evaluator verified that the TOE is able to encrypt the entire traffic into a TLS tunnel.

3. Security Assurance Requirement Assurance Activities

3.1 Development (ADV)

3.1.1 Basic Functional Specification (ADV_FSP.1)

3.1.1.1 Assurance Activity

There are no specific assurance activities associated with these SARs, except ensuring the information is provided. The functional specification documentation is provided to support the evaluation activities described in Section 5.1, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because there is insufficient interface information, then an adequate functional specification has not been provided.

The Assurance Activities identified above provided sufficient information to determine the appropriate content for the TSS section and to perform the assurance activities. Since these are directly associated with the SFRs, and are implicitly already done, no additional documentation or analysis is necessary.

3.2 Guidance Documents (AGD)

3.2.1 Operational User Guidance (AGD_OPE.1)

3.2.1.1 Assurance Activity

Some of the contents of the operational guidance will be verified by the assurance activities in Section 5.1 and evaluation of the TOE according to the [CEM]. The following additional information is also required.

If cryptographic functions are provided by the TOE, the operational guidance shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

The documentation must describe the process for verifying updates to the TOE by verifying a digital signature – this may be done by the TOE or the underlying platform.

The evaluator shall verify that this process includes the following steps:

Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).

Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature. The TOE will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.

Section “Software Download and Installation”, subsection “Installation” of [CCECG] provides instructions to the administrator to configure FIPS-CC mode for each platform version of the TOE. It also states this process ensures the TOE uses only the approved cryptographic module (OpenSSL FIPS canister engine) to perform all cryptographic functions and operations.

The guidance provided by [CCECG] describes the process to check for, download, and install updates to the TOE. The description covers how to obtain the update and make it accessible to the TOE and how to initiate the update process. It states that during the installation process the TOE automatically performs a digital signature verification check to verify the integrity of the update. If the check fails, the TOE displays this error message: "The Installation Failed: The Installer encountered an error that caused the installation to fail". More details on the error can be found in the install.log. If the process succeeds and the TOE is updated, this is also logged in the install.log. Refer to section "TOE Operation", subsection "Checking for Updates" of [CCECG].

Section "Introduction", subsection "Scope of Evaluation" of [CCECG] states the scope of evaluation of the TOE covers only the security functionality specified by the Protection Profile for Application Software and Functional Package for Transport Layer Security, and outlined in [ST]. This security functionality includes TLS trusted channels, X509 authentication, certificate validation, and signature checking.

3.2.2 Preparative Procedures (AGD_PRE.1)

3.2.2.1 Assurance Activity

As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.

The TOE in its evaluated configuration is supported on five platforms (Android, Windows, iOS, macOS, Linux) that are adequately addressed in the guidance documentation. Section "Software Download and Installation", subsection "Installation" of [CCECG] provides the instructions necessary to install and configure the TOE on each supported platform.

3.3 Tests (ATE)

3.3.1 Independent Testing – Conformance (ATE_IND.1)

3.3.1.1 Assurance Activity

The evaluator shall prepare a test plan and report documenting the testing aspects of the system, including any application crashes during testing. The evaluator shall determine the root cause of any application crashes and include that information in the report. The test plan covers all of the testing actions contained in the [CEM] and the body of this PP's Assurance Activities.

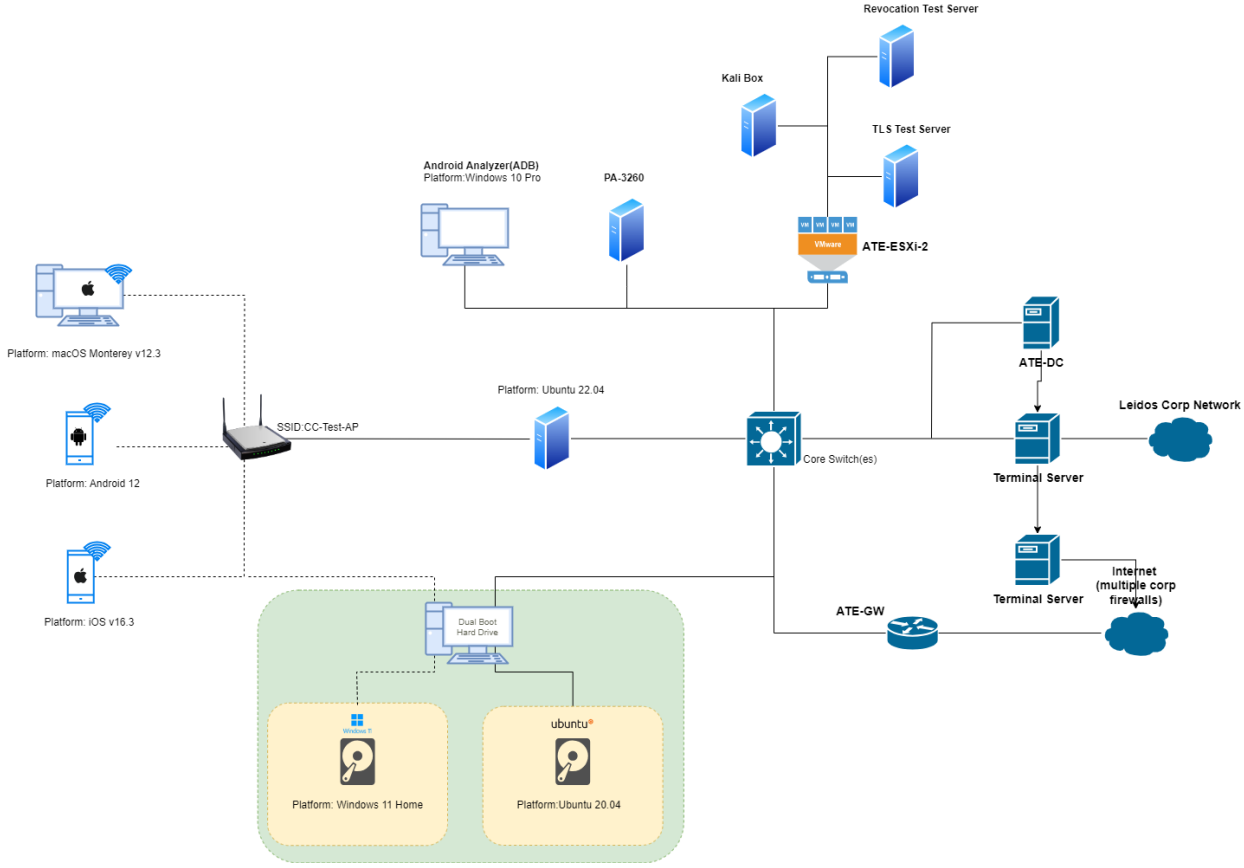
While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluator must document in the test plan that each applicable testing requirement in the ST is covered. The test plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary. The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform.

This also includes the configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by this PP and used by the cryptographic protocols being evaluated (IPsec, TLS, SSH). The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results.

The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a “fail” and “pass” result (and the supporting details), and not just the “pass” result.

The evaluation team tested the TOE at Leidos’s Columbia, MD location from May 2023 to July 2023. The procedures and results of this testing are available in the DTR document.

The following figure depicts the test configuration used by the evaluation team to test the TOE on each of its supported platforms.



The evaluation team used following components to create the test configuration:

TOE Platforms:

- HP Envy x360 2-in-1 Laptop, configured for Dual Boot
  - Windows 11
  - Ubuntu 20.04
- iPhone 12 mini
  - iOS v16.3
- MacBook Pro (14-inch, 2021)
  - macOS Monterey v12.3
- Galaxy S21 Ultra 5G
  - Android v12.3, Hostname: Galaxy S21 Ultra 5G

Test Configuration Components:

- Palo Alto Networks PA-3260 Firewall—hosts GlobalProtect Portal and Gateway
- Router—connects between wireless and wired networks
  - OpenSSL 3.0.2
- Wireless AP—allows wireless mobile devices to connect to test network
- Android Analyzer (ADB)
  - Android Studio (Electric Eel 2022.1.1)

- ATE-GW—Main router/gateway
- ATE-DC—Main Domain Controller (DC) for Test environment/DNS server
- ATE-ESXi-2—Virtualization server hosting:
  - Revocation Test Server--hosts TLS/OCSP test tools
    - OpenSSL 1.1.1
    - Wireshark 2.6.10
  - TLS Test Server—hosts TLS test tools
    - Proprietary Python TLS test tools
    - OpenSSL 1.1.1
    - Wireshark 2.6.10
  - Kali Box—hosts testing tools
    - SSLyze v2.0.6
    - OpenSSL 1.1.1
- Terminal Server—provides tester access to the Test Environment from corporate network.

3.4 Vulnerability Assessment (AVA)

3.4.1 Vulnerability Survey (AVA_VAN.1)

3.4.1.1 Assurance Activity

The evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to find vulnerabilities that have been found in similar applications with a particular focus on network protocols the application uses and document formats it parses.

The evaluator documents the sources consulted and the vulnerabilities found in the report.

For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.

For Windows, Linux, macOS and Solaris: The evaluator shall also run a virus scanner with the most current virus definitions against the application files and verify that no files are flagged as malicious.

The evaluation team performed a search of the following public vulnerability databases:

- National Vulnerability Database (NVD) (https://nvd.nist.gov/vuln/search)
- Palo Alto Networks Security Advisories (https://security.paloaltonetworks.com/)

The evaluation team performed searches on 28 August 2023 using the following search terms:

- “Palo Alto Networks” – TOE vendor
- “GlobalProtect” – TOE name
- “VPN Client” – TOE application type
- “OpenSSL 1.1.1” – Third-party library included with TOE
- “OESIS” – third-party library included with TOE.

The evaluation team did not identify any vulnerabilities in the TOE.

The evaluation team scanned the TOE's executable files using a corporate provided virus scan software (Microsoft Windows Defender) and verified that no virus signatures were detected.

The evaluation team determined that no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.

3.5 Life-Cycle Support (ALC)

3.5.1 Labeling of the TOE (ALC_CMC.1)

3.5.1.1 Assurance Activity

The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

Section 1.1 of [ST] ("Security Target, TOE and CC Identification") includes the TOE identification. The TOE is identified in terms of the software included in the evaluated configuration. This consists of the following platform-specific images:

- Windows 11—GlobalProtect64-6.0.7.msi
- macOS 12—GlobalProtect-6.0.7.pkg
- Android 12—global-protect-6.0.7-signed.apk
- iOS 16—GlobalProtect App downloaded from iTunes Store
- Linux Ubuntu 20.04—PanGPLinux-6.0.7.tgz.

This is consistent with the version number of the TOE identified by the TOE samples received for testing and in section "Introduction", subsection "Scope of Evaluation" of [CCECG].

3.5.2 TOE Coverage (ALC_CMS.1)

3.5.2.1 Assurance Activity

The "evaluation evidence required by the SARs" in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the assurance activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component. Life-cycle support is targeted aspects of the developer's life-cycle and instructions to providers of applications for the developer's devices, rather than an in-depth examination of the TSF manufacturer's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it's a reflection on the information to be made available for evaluation.

The evaluator shall ensure that the developer has identified (in guidance documentation for application developers concerning the targeted platform) one or more development environments appropriate for use in developing applications for the developer's platform. For each of these development environments, the developer shall provide information on how to configure the environment to ensure that buffer overflow protection mechanisms in the environment(s) are invoked (e.g., compiler flags). The evaluator shall ensure that this documentation also includes an indication of whether such protections are on by default, or have to be specifically enabled. The evaluator shall ensure that the TSF is uniquely identified (with respect to other products from the TSF vendor), and that documentation provided by the developer in association with the requirements in the ST is associated with the TSF using this unique identification.

As described in Section 3.5.1 above, the evaluator confirmed the TOE is labelled with unique software version identifiers. Section 6.6 of [ST] ("Protection of the TSF") describes how the TOE uses platform security features and APIs. This includes data execution protection and stack-based buffer overflow protection.

3.5.3 Timely Security Update (ALC_TSU_EXT.1)

3.5.3.1 Assurance Activity

The evaluator shall verify that the TSS contains a description of the timely security update process used by the developer to create and deploy security updates. The evaluator shall verify that this description addresses the entire application. The evaluator shall also verify that, in addition to the TOE developer's process, any third-party processes are also addressed in the description. The evaluator shall also verify that each mechanism for deployment of security updates is described.

The evaluator shall verify that, for each deployment mechanism described for the update process, the TSS lists a time between public disclosure of a vulnerability and public availability of the security update to the TOE patching this vulnerability, to include any third-party or carrier delays in deployment. The evaluator shall verify that this time is expressed in a number or range of days.

The evaluator shall verify that this description includes the publicly available mechanisms (including either an email address or website) for reporting security issues related to the TOE. The evaluator shall verify that the description of this mechanism includes a method for protecting the report either using a public key for encrypting email or a trusted channel for a website.

Section 6.6 of [ST] ("Protection of the TSF") describes the vendor's timely security update process.

Palo Alto Networks regularly issues maintenance releases for the TOE once a major release is made available to the public. Maintenance releases can include bug fixes to improve product features and to address any security vulnerabilities that may have been identified in previous versions. When a new version is available, Palo Alto Networks notifies users via an email with the specific version published. These versions are also displayed on Palo Alto Networks' Customer Support page (https://support.paloaltonetworks.com). Palo Alto Networks provides an updated version of the product on a regular basis to customers.

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.

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.

Palo Alto Networks provides timely security updates to its customers. Depending on the CVSS (Common Vulnerability Scoring System), the security updates can be provided as quickly as 2 weeks via a security hotfix release.