IBM MaaS360 2.106.500.016 Cloud Extender
Assurance Activity Report

Version:	1.2
Date:	2022-09-12
Status:	RELEASED
Classification:	Public
Filename:	VID11113_SER_AAR_IBM_MaaS360_CE_v1.2
Product:	IBM MaaS360 2.106.500.016 Cloud Extender
Sponsor:	IBM Corp.
Evaluation Facility:	atsec information security corporation
Validation ID:	11113
Validation Body:	NIAP CCEVS
Author(s):	Randy Baker
Quality Assurance:	King Ables

atsec information security corporation
9130 Jollyville Road, Suite 260
Austin, TX 78759

Phone: +1 512-615-7300
www.atsec.com

Classification Note

Public Information (public)

This classification level is for information that may be made available to the general public. No specific security procedures are required to protect the confidentiality of this information. Information classified “public” may be freely distributed to anyone inside or outside of atsec.

Information with this classification shall be clearly marked “public”, except that it is not required to mark “public” on printed marketing material obviously intended for publication.

Revision History

Version	Date	Author(s)	Changes to Previous Revision	Application Notes
1.0	2022-08-15	Randy Baker	First version	
1.1	2022-09-06	Randy Baker	Address validator comments.	
1.2	2022-09-12	King Ables	Rebuild for AVA update.	

Table of Contents

List of Tables

1 Evaluation Basis and Documents

This evaluation is based on the "Common Criteria for Information Technology Security Evaluation" Version 3.1 Revision 5 [CC], the "Common Methodology for Information Technology Security Evaluation" [CEM] and the additional assurance activities defined in the following:

- [ASPPv1.3]: Protection Profile for Application Software, Version 1.3, dated 2019-03-01
- [TLSPKGv1.1]: Functional Package for TLS, Version 1.1, dated 2019-03-01

This evaluation claims Exact Compliance with the above PP and Extended Package.

The following scheme documents and interpretations have been considered:

- [CCEVS-TD0416]: "Correction to FCS_RBG_EXT.1 Test Activity", version as of 2019-04-24.
- [CCEVS-TD0427]: "Reliable Time Source", version as of 2019-06-11.
- [CCEVS-TD0434]: "Windows Desktop Applications Test", version as of 2019-07-22.
- [CCEVS-TD0435]: "Alternative to SELinux for FPT_AEX_EXT.1.3", version as of 2019-07-26.
- [CCEVS-TD0437]: "Supported Configuration Mechanism", version as of 2019-07-23.
- [CCEVS-TD0442]: "Updated TLS Ciphersuites for TLS Package", version as of 2019-08-21.
- [CCEVS-TD0445]: "User Modifiable File Definition", version as of 2019-10-09.
- [CCEVS-TD0465]: "Configuration Storage for .NET Apps", version as of 2019-11-08.
- [CCEVS-TD0469]: "Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1", version as of 2019-11-20.
- [CCEVS-TD0495]: "FIA_X509_EXT.1.2 Test Clarification", version as of 2020-01-29.
- [CCEVS-TD0498]: "Application Software PP Security Objectives and Requirements Rationale", version as of 2020-01-31.
- [CCEVS-TD0499]: "Testing with pinned certificates", version as of 2020-02-04.
- [CCEVS-TD0510]: "Obtaining random bytes for iOS/macOS", version as of 2020-03-03.
- [CCEVS-TD0513]: "CA Certificate loading", version as of 2020-05-26.
- [CCEVS-TD0515]: "Use Android APK manifest in test", version as of 2020-06-08.
- [CCEVS-TD0519]: "Linux symbolic links and FMT_CFG_EXT.1", version as of 2020-06-18.
- [CCEVS-TD0543]: "FMT_MEC_EXT.1 evaluation activity update", version as of 2020-09-15.
- [CCEVS-TD0544]: "Alternative testing methods for FPT_AEX_EXT.1.1", version as of 2020-09-15.
- [CCEVS-TD0548]: "Integrity for installation tests in AppSW PP 1.3", version as of 2020-09-30.
- [CCEVS-TD0554]: "iOS/iPadOS/Android AppSW Virus Scan", version as of 2020-10-30.
- [CCEVS-TD0561]: "Signature verification update", version as of 2021-01-15.
- [CCEVS-TD0582]: "PP-Configuration for Application Software and Virtual Private Network (VPN) Clients now allowed", version as of 2021-04-16.
- [CCEVS-TD0588]: "Session Resumption Support in TLS package", version as of 2021-05-12.
- [CCEVS-TD0598]: "Expanded AES Modes in FCS_COP for App PP", version as of 2021-08-03.
- [CCEVS-TD0600]: "Conformance claim sections updated to allow for MOD_VPNC_V2.3", version as of 2021-08-10.

- [CCEVS-TD0601]: "X.509 SFR Applicability in App PP", version as of 2021-09-22.

2 Evaluation Results

The evaluator work units have been performed, including: evaluator actions and analysis explicitly stated in the CEM; evaluator actions implicitly derived from developer action elements described in the CC Part 3; and evaluator confirmation that requirements for content and presentation of evidence elements described in the CC Part 3 have been met.

The evaluation was performed by informal analysis of the evidence provided by the sponsor.

2.1 CAVP Summary

The following table summarizes the Cryptographic Algorithm Validation Program (CAVP) certificates that apply to the TOE, including specific standards, options, and implementations for each algorithm of each SFR, and the applicable CAVP certificate for each.

The TOE's CAVP certificate Operational Environment (OE) is the following.

- Microsoft Windows Server 2019 Standard version 1809 (x64):
  - Intel SkyLake Gold 5118

Table 1: SFRs to CAVP certificates for IBM MaaS360 Cloud Extender

SFR	Algorithm	Modes / Other	Implementation	CAVP
FCS_CKM.1(1)	RSA KeyGen [FIPS 186-4]	Modulo: 2048, 3072, 4096	OpenSSL	A2519
	ECDSA KeyGen [FIPS 186-4]	P-256, P-384	OpenSSL	A2519
			CNG/ SymCrypt	C211
	ECDSA KeyVer [FIPS 186-4]	P-256, P-384	OpenSSL	A2519
			CNG/ SymCrypt	C211
FCS_CKM.2	ECC Key Establishment (KAS-ECC Component) Ephemeral Unified [SP800-56A]	P-256, P-384 SHA2-256, SHA2-384	OpenSSL	A2519
	ECC Key Establishment (KAS-ECC) Ephemeral Unified [SP800-56A]		CNG/ SymCrypt	C211

SFR	Algorithm	Modes / Other	Implementation	CAVP
FCS_COP.1(1)	AES	CBC[1] 128-bit, 256-bit [SP800-38A] (CBC)	OpenSSL	A2519
			CNG/ SymCrypt	C211
		GCM[2] 128-bit [SP800-38D] (GCM)	OpenSSL	A2519
			CNG/ SymCrypt	C211
FCS_COP.1(2)	SHS Byte-oriented mode [FIPS 180-4]	SHA-1, SHA2-256, SHA2-384, SHA2-512	OpenSSL	A2519
			CNG/ SymCrypt	C211
FCS_COP.1(3)	RSA SigGen PKCS 1.5 [FIPS 186-4]	Modulo: 2048, 3072, 4096 using SHA2-256, SHA2-384	OpenSSL	A2519
		Modulo: 2048, 3072 using SHA2-256, SHA2-384	CNG/ SymCrypt	C211
	RSA SigVer PKCS 1.5 [FIPS 186-4]	Modulo: 2048, 3072, 4096 using SHA-1, SHA2-256, SHA2-384	OpenSSL	A2519
		Modulo: 2048, 3072 using	CNG/ SymCrypt	C211

[1] Note: AES-CBC with 256-bit keys is required for the CTR_DRBG method used by the SP800-90A DRBG implemented in the OpenSSL for the IBM MaaS360 Cloud Extender cryptographic module.
[2] Note: AES-GCM with 128-bit keys is required for the TLS cipher suites supported by the TOE.

SFR	Algorithm	Modes / Other	Implementation	CAVP
		SHA-1, SHA2-256, SHA2-384		
FCS_COP.1(4)	HMAC Byte-oriented mode [FIPS 198-1]	HMAC-SHA2-256	OpenSSL	A2519
			CNG/ SymCrypt	C211
FCS_RBG_EXT.2	CTR_DRBG(AES) [SP800-90A]	AES-256 Entropy Input: 256 Nonce: 128	OpenSSL	A2519
			CNG/ SymCrypt	C211

2.2 Test configuration

Platform configuration and operational environment conforms to that described in [ST] and [CC-CFG]. The test platform is a Dell PowerEdge R740 with Intel Xeon Gold 5118 processor (Skylake microarchitecture) as described in section 1.5.2.1.1 of [ST]. The TOE is distributed in a Common Criteria-specific package consisting of a Microsoft Windows installer to be installed on the hardware platform. The TOE is installed and configured as specified in [CC-CFG] and [ADM_GUIDE]. The TOE is in the evaluated configuration at the start of each test.

2.3 Security Functional Requirements

2.3.1 Cryptographic support (FCS)

2.3.1.1 Cryptographic Asymmetric Key Generation (FCS_CKM.1(1))

TSS Assurance Activities

Assurance Activity AA-FCS_CKM.1-1-ASE-01

> *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.*
>
> *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.*

Summary

Table 15 in [ST] lists the key sizes for all asymmetric keys types specified in FCS_CKM.1 and generated by both the platform and the TOE. Section 7.1.1.1, "Cryptographic Algorithms," also states that OpenSSL is used for CSRs and TLS whereas the platform-provided cryptographic functionality is used for TLS. Table 15 also states that Ephemeral asymmetric key generation is used for both the TOE (OpenSSL) and platform-provided cryptographic functionality (TLS).

Guidance Assurance Activities

Assurance Activity AA-FCS_CKM.1-1-AGD-01

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.

Summary

The evaluator verified that sections 2.1 to 2.3 of [CC-CFG] specify how to restrict the algorithms, key sizes, and protocol version for the TOE, as defined in the PP, using the Windows key registry.

The TOE itself does not support the possibility to configure the key schemes and key sizes. Whenever the TOE is installed and configured in the evaluated configuration by the administrator, the user cannot modify the key generation schemes and the key sizes set by the administrator.

Test Assurance Activities

Assurance Activity AA-FCS_CKM.1-1-ATE-01

*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.

Key Generation for FIPS PUB 186-4 RSA Schemes

The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d. Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include:

1. *Random Primes:*
   - *Provable primes*
   - *Probable primes*
2. *Primes with Conditions:*
   - *Primes p1, p2, q1,q2, p and q shall all be provable primes*
   - *Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes*
   - *Primes p1, p2, q1,q2, p and q shall all be probable primes*

To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation.

If possible, the Random Probable primes method should also be verified against a known good implementation as described above. Otherwise, the evaluator shall have the TSF generate 10 keys pairs for each supported key length nlen and verify:

- $n = p \cdot q$,
- *p and q are probably prime according to Miller-Rabin tests,*
- $GCD(p-1,e) = 1$,
- $GCD(q-1,e) = 1$,
- $2^{16} \leq e \leq 2^{256}$ *and e is an odd integer,*
- $|p-q| > 2^{nlen/2 - 100}$,
- $p \geq 2^{nlen/2 -1/2}$,
- $q \geq 2^{nlen/2 -1/2}$,
- $2^{(nlen/2)} < d < LCM(p-1,q-1)$,
- $e \cdot d = 1 \bmod LCM(p-1,q-1)$.

Key Generation for Elliptic Curve Cryptography (ECC)

FIPS 186-4 ECC Key Generation Test For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation.

FIPS 186-4 Public Key Verification (PKV) Test For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values.

Key Generation for Finite-Field Cryptography (FFC)

The evaluator shall verify the implementation of the Parameters Generation and the Key Generation for FFC by the TOE using the Parameter Generation and Key Generation test. This test verifies the ability of the TSF to correctly produce values for the field prime p, the cryptographic prime q (dividing p-1), the cryptographic group generator g, and the calculation of the private key x and public key y. The Parameter generation specifies 2 ways (or methods) to generate the cryptographic prime q and the field prime p:

Cryptographic and Field Primes:

- *Primes q and p shall both be provable primes*
- *Primes q and field prime p shall both be probable primes*

and two ways to generate the cryptographic group generator g:
Cryptographic Group Generator:

- *Generator g constructed through a verifiable process*
- *Generator g constructed through an unverifiable process.*

The Key generation specifies 2 ways to generate the private key x: Private Key:

- *len(q) bit output of RBG where 1 ≤x ≤ q-1*
- *len(q) + 64 bit output of RBG, followed by a mod q-1 operation where 1≤ x≤q-1.*

The security strength of the RBG must be at least that of the security offered by the FFC parameter set. To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator must seed the TSF parameter generation routine with sufficient data to deterministically generate the parameter set. For each key length supported, the evaluator shall have the TSF generate 25 parameter sets and key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. Verification must also confirm

- *g ≠ 0,1*
- *q divides p-1*
- *g^q mod p = 1*
- *g^x mod p = y*

for each FFC parameter set and key pair.

Diffie-Hellman Group 14 and FFC Schemes using "safe-prime" groups

Testing for FFC Schemes using Diffie-Hellman group 14 and/or safe-prime groups is done as part of testing in CKM.2.1.

Summary

The evaluator confirmed that CAVS testing of all key generation operations was performed in accordance with the CAVP test procedures. The results have been validated by the CAVP and the certificate information is provided in section 2.1, "CAVP Summary," in the Assurance Activity Report (AAR).

2.3.1.2 Cryptographic Key Establishment (FCS_CKM.2)

TSS Assurance Activities

Assurance Activity AA-FCS_CKM.2-ASE-01

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

Summary

The Elliptic curve-based key establishment scheme listed in FCS_CKM.2 corresponds to the ECC key generation scheme listed in FCS_CKM.1. Only one key establishment scheme is claimed. The evaluator notes that TLS does not use a ciphersuite where RSA keys are exchanged.

Guidance Assurance Activities

Assurance Activity AA-FCS_CKM.2-AGD-01

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

Summary

In [CC-CFG], section 2.1, *Make TLS 1.2 the System Default on Windows Server 2019*, describes how to configure TLS 1.2 for for TOE Windows platform.

The evaluator reviewed section 2.1 and determined that some configuration is required to configure the functionality for the required key establishment schemes. More specifically, section 2.1 describes how to restrict the TOE to only use TLS v1.2 with the following cipher suite.

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Test Assurance Activities

Assurance Activity AA-FCS_CKM.2-ATE-01

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

The evaluator shall verify a TOE's implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the DLC primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag.

Function Test

The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested.

The evaluator shall obtain the DKM, the corresponding TOE's public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information (OtherInfo) and TOE id fields.

If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret.

The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values.

If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm.

Validity Test

The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOE's public/private key pairs, MACTag, and any inputs used in the KDF, such as the OtherInfo and TOE id fields.

The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the OtherInfo field, the data to be MACed, or the generated MACTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the TOE's static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass).

The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOE's results with the results using a known good implementation verifying that the TOE detects these errors.

SP800-56B Key Establishment Schemes

The evaluator shall verify that the TSS describes whether the TOE acts as a sender, a recipient, or both for RSA-based key establishment schemes.

If the TOE acts as a sender, the following evaluation activity shall be performed to ensure the proper operation of every TOE supported combination of RSA-based key establishment scheme:

> *To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each combination of supported key establishment scheme and its options (with or without key confirmation if supported, for each supported key confirmation MAC function if key confirmation is supported, and for each supported mask generation function if KTS-OAEP is supported), the tester shall generate 10 sets of test vectors. Each test vector shall include the RSA public key, the plaintext keying material, any additional input parameters if applicable, the MacKey and MacTag if key confirmation is incorporated, and the outputted ciphertext. For each test vector, the evaluator shall perform a key establishment encryption operation on the TOE with the same inputs (in cases where key confirmation is incorporated, the test shall use the MacKey from the test vector instead of the randomly generated MacKey used in normal operation) and ensure that the outputted ciphertext is equivalent to the ciphertext in the test vector.*

If the TOE acts as a receiver, the following evaluation activities shall be performed to ensure the proper operation of every TOE supported combination of RSA-based key establishment scheme:

> *To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each combination of supported key establishment scheme and its options (with our without key confirmation if supported, for each supported key confirmation MAC function if key confirmation is supported, and for each supported mask generation function if KTS-OAEP is supported), the tester shall generate 10 sets of test vectors. Each test vector shall include the RSA private key, the plaintext keying material (KeyData), any additional input parameters if applicable, the MacTag in cases where key confirmation is incorporated, and the outputted ciphertext. For each test vector, the evaluator shall perform the key establishment decryption operation on the TOE and ensure that the outputted plaintext keying material (KeyData) is equivalent to the plaintext keying material in the test vector. In cases where key confirmation is incorporated, the evaluator shall perform the key confirmation steps and ensure that the outputted MacTag is equivalent to the MacTag in the test vector.*

The evaluator shall ensure that the TSS describes how the TOE handles decryption errors. In accordance with NIST Special Publication 800-56B, the TOE must not reveal the particular error that occurred, either through the contents of any outputted or logged error message or through timing variations. If KTS-OAEP is supported, the evaluator shall create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 800-56B section 7.2.2.3, ensure that each decryption attempt results in an error, and ensure that any

outputted or logged error message is identical for each. If KTS-KEM-KWS is supported, the evaluator shall create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 800-56B section 7.2.3.3, ensure that each decryption attempt results in an error, and ensure that any outputted or logged error message is identical for each.

RSA-based key establishment

The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_DIT_EXT.1 that uses RSAES-PKCS1-v1_5.

Diffie-Hellman Group 14

The evaluator shall verify the correctness of the TSF's implementation of Diffie-Hellman group 14 by using a known good implementation for each protocol selected in FTP_DIT_EXT.1 that uses Diffie-Hellman group 14.

FFC Schemes using "safe-prime" groups

The evaluator shall verify the correctness of the TSF's implementation of safe-prime groups by using a known good implementation for each protocol selected in FTP_DIT_EXT.1 that uses safe-prime groups. This test must be performed for each safe-prime group that each protocol uses.

Summary

The evaluator confirmed that CAVS testing of all key generation operations was performed in accordance with the CAVP test procedures. The results have been validated by the CAVP and the certificate information is provided in section 2.1, "CAVP Summary," in the Assurance Activity Report (AAR).

2.3.1.3 Cryptographic Key Generation Services (FCS_CKM_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FCS_CKM_EXT.1-ASE-01

*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.*

Summary

[ST] selects both "invoke platform-provided functionality for asymmetric key generation" and "implement asymmetric key generation" in FCS_CKM_EXT.1. The associated assurance activities are performed below.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.1.4 Cryptographic Operation - Encryption/Decryption (FCS_COP.1(1))

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FCS_COP.1-1-AGD-01

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.

Summary

[CC-CFG], section 2.1, *Make TLS 1.2 the System Default on Windows Server 2019*, describes how to configure TLS 1.2 for the Windows platform. Section 2.2, *Enable FIPS and NIAP Modes and disable module updates*, describes how to configure TLS 1.2 for OpenSSL.

The evaluator reviewed sections 2.1 and 2.2 and determined that some configuration is required to configure the functionality for the required modes and key sizes. More specifically, section 2.1 describes how to restrict the TOE to only use TLS v1.2 with the following cipher suite.

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Test Assurance Activities

Assurance Activity AA-FCS_COP.1-1-ATE-01

The evaluator shall perform all of the following tests for each algorithm implemented by the TSF and used to satisfy the requirements of this PP:

AES-CBC Known Answer Tests

There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

- *KAT-1. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all- zeros key. To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption.*
- *KAT-2. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys. To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using an all-zero ciphertext value as input and AES-CBC decryption.*
- *KAT-3. To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. To test the decrypt functionality of AES-CBC, the evaluator shall supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. The ciphertext value in each pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key.*
- *KAT-4. To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256-bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128].*

To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption.

AES-CBC Multi-Block Message Test

The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 < i <= 10. The evaluator shall choose a key, an IV and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation. The evaluator shall also test the decrypt functionality for each mode by decrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation. AES-CBC Monte Carlo Tests The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3- tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:

```
# Input: PT, IV, Key
for i = 1 to 1000:
if i == 1:
CT[1] = AES-CBC-Encrypt(Key, IV, PT)
PT = IV
else:
CT[i] = AES-CBC-Encrypt(Key, PT)
PT = CT[i-1]
```

The ciphertext computed in the 1000th iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

The evaluator shall test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AES-CBC-Decrypt.

AES-GCM Monte Carlo Tests

The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths:

- *128 bit and 256 bit keys*
- *Two plaintext lengths. One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported.*
- *Three AAD lengths. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported.*
- *Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.*

The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known.

The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail.

The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

AES-XTS Tests

The evaluator shall test the encrypt functionality of XTS-AES for each combination of the following input parameter lengths:

256 bit (for AES-128) and 512 bit (for AES-256) keys

Three data unit (i.e., plaintext) lengths. One of the data unit lengths shall be a non-zero integer multiple of 128 bits, if supported. One of the data unit lengths shall be an integer multiple of 128 bits, if supported. The third data unit length shall be either the longest supported data unit length or 216 bits, whichever is smaller.

Using a set of 100 (key, plaintext and 128-bit random tweak value) 3-tuples and obtain the ciphertext that results from XTS-AES encrypt.

The evaluator may supply a data unit sequence number instead of the tweak value if the implementation supports it. The data unit sequence number is a base-10 number ranging between 0 and 255 that implementations convert to a tweak value internally.

The evaluator shall test the decrypt functionality of XTS-AES using the same test as for encrypt, replacing plaintext values with ciphertext values and XTS-AES encrypt with XTS-AES decrypt.

AES-CCM Tests [TD0598]

It is not recommended that evaluators use values obtained from static sources such as http://csrc.nist.gov/groups/STM/cavp/documents/mac/ccmtestvectors.zip or use values not generated expressly to exercise the AES-CCM implementation.

The evaluator shall test the generation-encryption and decryption-verification functionality of AES-CCM for the following input parameter and tag lengths:

Keys: All supported and selected key sizes (e.g., 128, 256 bits).

Associated Data: Two or three values for associated data length: The minimum (≥ 0 bytes) and maximum (≤ 32 bytes) supported associated data lengths, and 2^16 (65536) bytes, if supported.

Payload: Two values for payload length: The minimum (≥ 0 bytes) and maximum (≤ 32 bytes) supported payload lengths.

Nonces: All supported nonce lengths (7, 8, 9, 10, 11, 12, 13) in bytes.

Tag: All supported tag lengths (4, 6, 8, 10, 12, 14, 16) in bytes.

The testing for CCM consists of five tests. To determine correctness in each of the below tests, the evaluator shall compare the ciphertext with the result of encryption of the same inputs with a known good implementation.

Variable Associated Data Test

For each supported key size and associated data length, and any supported payload length, nonce length, and tag length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.

Variable Payload Test

For each supported key size and payload length, and any supported associated data length, nonce length, and tag length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.

Variable Nonce Test

For each supported key size and nonce length, and any supported associated data length, payload length, and tag length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.

Variable Tag Test

For each supported key size and tag length, and any supported associated data length, payload length, and nonce length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.

Decryption-Verification Process Test

To test the decryption-verification functionality of AES-CCM, for each combination of supported associated data length, payload length, nonce length, and tag length, the evaluator shall supply a key value and 15 sets of input plus ciphertext, and obtain the decrypted payload. Ten of the 15 input sets supplied should fail verification and five should pass.

AES-CTR Tests [TD0598]

Test 1*: Known Answer Tests (KATs)*

There are four Known Answer Tests (KATs) described below. For all KATs, the plaintext, IV, and ciphertext values shall be 128-bit blocks. The results from each test may either be obtained by the validator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

To test the encrypt functionality, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all zeros key, and the other five shall be encrypted with a 256-bit all zeros key. To test the decrypt functionality, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input.

To test the encrypt functionality, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from encryption of an all zeros plaintext using the given key value and an IV of all zeros. Five of the key values shall be 128-bit keys, and the other five shall be 256-bit keys. To test the decrypt functionality, the evaluator shall perform the same test as for encrypt, using an all zero ciphertext value as input.

To test the encrypt functionality, the evaluator shall supply the two sets of key values described below and obtain the ciphertext values that result from AES encryption of an all zeros plaintext using the given key values an an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second shall have 256 256-bit keys. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1, N]. To test the decrypt functionality, the evaluator shall supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value that results from decryption of the given ciphertext using the given key values and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit pairs. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros for i in [1, N]. The ciphertext value in each pair shall be the value that results in an all zeros plaintext when decrypted with its corresponding key.

To test the encrypt functionality, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from encryption of the given plaintext using a 128-bit key value of all zeros and using a 256 bit key value of all zeros, respectively, and an IV of all zeros. Plaintext value i in each set shall have the leftmost bits be ones and the rightmost 128-i bits be zeros, for i in [1, 128]. To test the decrypt functionality, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input.

***Test 2**: Multi-Block Message Test*

The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10. For each i the evaluator shall choose a key, IV, and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation. The evaluator shall also test the decrypt functionality by decrypting an i-block message where 1 less-than i less-than-or-equal to 10. For each i the evaluator shall choose a key and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key using a known good implementation.

***Test 3**: Monte-Carlo Test*

For AES-CTR mode perform the Monte Carlo Test for ECB Mode on the encryption engine of the counter mode implementation. There is no need to test the decryption engine.

The evaluator shall test the encrypt functionality using 200 plaintext/key pairs. 100 of these shall use 128 bit keys, and 100 of these shall use 256 bit keys. The plaintext values shall be 128-bit blocks. For each pair, 1000 iterations shall be run as follows:

For AES-ECB mode

 # Input: PT, Key

for i = 1 to 1000:

CT[i] = AES-ECB-Encrypt(Key, PT)

PT = CT[i]

The ciphertext computed in the 1000th iteration is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

Summary

The evaluator confirmed that CAVS testing of all encryption and decryption operations was performed in accordance with the CAVP test procedures. The results have been validated by the CAVP and the certificate information is provided in section 2.1, "CAVP Summary," in the Assurance Activity Report (AAR).

2.3.1.5 Cryptographic Operation - Hashing (FCS_COP.1(2))

TSS Assurance Activities

Assurance Activity AA-FCS_COP.1-2-ASE-01

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.

Summary

The hash functions listed in FCS_COP.1(2) are listed in Table 15 in TSS and associated with the Pseudorandom function (PRF) and Digital Signature Generation and Verification.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_COP.1-2-ATE-01

The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF hashes only messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented testmacs. The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP.

The following tests require the developer to provide access to a test application that provides the evaluator with tools that are typically not found in the production application.

- ***Test 1:*** *Short Messages Test - Bit oriented Mode The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*
- ***Test 2:*** *Short Messages Test - Byte oriented Mode The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*
- ***Test 3:*** *Selected Long Messages Test - Bit oriented Mode The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*
- ***Test 4:*** *Selected Long Messages Test - Byte oriented Mode The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.*
- ***Test 5:*** *Pseudorandomly Generated Messages Test This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.*

Summary

The evaluator confirmed that CAVS testing of all hash operations was performed in accordance with the CAVP test procedures. The results have been validated by the CAVP and the certificate information is provided in section 2.1, "CAVP Summary," in the Assurance Activity Report (AAR).

2.3.1.6 Cryptographic Operation - Signing (FCS_COP.1(3))

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_COP.1-3-ATE-01

The following tests require the developer to provide access to a test application that provides the evaluator with tools that are typically not found in the production application.

ECDSA Algorithm Tests

- ***Test 1:*** *ECDSA FIPS 186-4 Signature Generation Test. For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation.*
- ***Test 2:*** *ECDSA FIPS 186-4 Signature Verification Test. For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.*

RSA Signature Algorithm Tests

- ***Test 1:*** *Signature Generation Test. The evaluator shall verify the implementation of RSA Signature Generation by the TOE using the Signature Generation Test. To conduct this test the evaluator must generate or obtain 10 messages from a trusted reference implementation for each modulus size/SHA combination supported by the TSF. The evaluator shall have the TOE use their private key and modulus value to sign these messages. The evaluator shall verify the correctness of the TSF's signature using a known good implementation and the associated public keys to verify the signatures.*
- ***Test 2:*** *Signature Verification Test. The evaluator shall perform the Signature Verification test to verify the ability of the TOE to recognize another party's valid and invalid signatures. The evaluator shall inject errors into the test vectors produced during the Signature Verification Test by introducing errors in some of the public keys, e, messages, IR format, and/or signatures. The TOE attempts to verify the signatures and returns success or failure.*

Summary

The evaluator confirmed that CAVS testing of all signing operations was performed in accordance with the CAVP test procedures. The results have been validated by the CAVP and the certificate information is provided in section 2.1, "CAVP Summary," in the Assurance Activity Report (AAR).

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

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_COP.1-4-ATE-01

For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and IV using a known-good implementation.

Summary

The evaluator confirmed that CAVS testing of all keyed-hash message authentication operations was performed in accordance with the CAVP test procedures. The results have been validated by the CAVP and the certificate information is provided in section 2.1, "CAVP Summary," in the Assurance Activity Report (AAR).

2.3.1.8 HTTPS Protocol (FCS_HTTPS_EXT.1/Client)

FCS_HTTPS_EXT.1.1-cli

TSS Assurance Activities

Assurance Activity AA-FCS_HTTPS_EXT.1.1-CLI-ASE-01

[TD0601] The evaluator shall examine the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818.

Summary

[ST] section 7.1.1.4, "HTTPS and TLS Protocols," explicitly states HTTPS is implemented as described in RFC2818.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_HTTPS_EXT.1.1-CLI-ATE-01

[TD0601] The evaluator shall attempt to establish an HTTPS connection with a webserver, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as TLS or HTTPS.

Summary

The evaluator successfully established an HTTPS connection to a test web server and verified using wireshark that the traffic was identified as TLS or HTTPS.

FCS_HTTPS_EXT.1.2-cli

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_HTTPS_EXT.1.2-CLI-ATE-01

[TD0601] Other tests are performed in conjunction with the TLS package.

Summary

The evaluator performed all testing required by FCS_TLSC_EXT.1 for HTTPS connections.

FCS_HTTPS_EXT.1.3-cli

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_HTTPS_EXT.1.3-CLI-ATE-01

[TD0601] Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1, and the evaluator shall perform the following test:

- ***Test 1:*** *The evaluator shall demonstrate that using a certificate without a valid certification path results in the selected action in the SFR. If "notify the user" isselected in the SFR, then the evaluator shall also determine that the user isnotified of the certificate validation failure. Using the administrative guidance, the evaluator shall then load a certificate or certificates to the Trust Anchor Database needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that again, using a certificate without a valid certification path results in the selected action in the SFR, and if "notify the user" was selected in the SFR, the user is notified of the validation failure.*

Summary

Certificate validation for HTTPS is performed in conjunction with Test 1 for FIA_X509_EXT.1.1.

2.3.1.9 Random Bit Generation Services (FCS_RBG_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FCS_RBG_EXT.1-ASE-01

*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.

Summary

FCS_RBG_EXT.1 in [ST] selects "implement DRBG functionality." The evaluator verified that FCS_RBG_EXT.2 is also claimed in [ST].

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_RBG_EXT.1-ATE-01

*[TD0416] 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.

The following are the per-platform list of acceptable APIs:

***For Android:** The evaluator shall verify that the application uses at least one of javax.crypto.KeyGenerator class or the java.security.SecureRandom class or /dev/random or /dev/urandom.*

***For Windows:** The evaluator shall verify that rand_s, RtlGenRandom, BCryptGenRandom, or CryptGenRandom API is used for classic desktop applications. The evaluator shall verify the application uses the RNGCryptoServiceProvider class or derives a class from System.Security.Cryptography.RandomNumberGenerator API for Windows Universal Applications. It is only required that the API is called/invoked, there is no requirement that the API be used directly. In future versions of this document, CryptGenRandom may be removed as an option as it is no longer the preferred API per vendor documentation.*

***For iOS:** [TD0510] The evaluator shall verify that the application invokes either SecRandomCopyBytes, CCRandomGenerateBytes or CCRandomCopyBytes, or uses /dev/random directly to acquire random.*

***For Linux:** The evaluator shall verify that the application collects random from /dev/random or /dev/urandom.*

***For Solaris:** The evaluator shall verify that the application collects random from /dev/random.*

***For macOS:** [TD0510] The evaluator shall verify that the application invokes either CCRandomGenerateBytes or CCRandomCopyBytes, or collects random from /dev/random.*

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).

Summary

The TOE implements random bit generation functionality using AES_CTR DRBG in OpenSSL. Entropy is gathered from a Windows platform as a seed using the BCryptGenRandom API.

2.3.1.10 Random Bit Generation from Application (FCS_RBG_EXT.2)

FCS_RBG_EXT.2.1

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_RBG_EXT.2.1-ATE-01

The evaluator shall perform the following tests, depending on the standard to which the RBG conforms.

Implementations Conforming to FIPS 140-2 Annex C.

The reference for the tests contained in this section is The Random Number Generator Validation System (RNGVS). The evaluators shall conduct the following two tests. Note that the "expected values" are produced by a reference implementation of the algorithm that is known to be correct. Proof of correctness is left to each Scheme.

- ***Test 1:*** *The evaluators shall perform a Variable Seed Test. The evaluators shall provide a set of 128 (Seed, DT) pairs to the TSF RBG function, each 128 bits. The evaluators shall also provide a key (of the length appropriate to the AES algorithm) that is constant for all 128 (Seed, DT) pairs. The DT value is incremented by 1 for each set. The seed values shall have no repeats within the set. The evaluators ensure that the values returned by the TSF match the expected values.*
- ***Test 2:*** *The evaluators shall perform a Monte Carlo Test. For this test, they supply an initial Seed and DT value to the TSF RBG function; each of these is 128 bits. The evaluators shall also provide a key (of the length appropriate to the AES algorithm) that is constant throughout the test. The evaluators then invoke the TSF RBG 10,000 times, with the DT value being incremented by 1 on each iteration, and the new seed for the subsequent iteration produced as specified in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section E.3. The evaluators ensure that the 10,000th value produced matches the expected value.*

Implementations Conforming to NIST Special Publication 800-90A

Test 1:

The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality.

If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90A).

If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

Entropy input: *the length of the entropy input value must equal the seed length.*

Nonce: *If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length.*

Personalization string: *The length of the personalization string must be less then or equal to seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.*

Additional input: *the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.*

Summary

The evaluator confirmed that CAVS testing of all RNG operations was performed in accordance with the CAVP test procedures. The results have been validated by the CAVP and the certificate information is provided in section 2.1, "CAVP Summary," in the Assurance Activity Report (AAR).

FCS_RBG_EXT.2.2

TSS Assurance Activities

Assurance Activity AA-FCS_RBG_EXT.2.2-ASE-01

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

Summary

The evaluator examined [EAR] provided by the developer. According to Appendix D of [ASPPv1.3], EAR must describe the following aspects of the TOE's entropy:

- Design description
- Operation of the entropy source
- Entropy justification
- Operating conditions
- Health testing

[EAR] section 1 states the TOE uses two cryptographic service providers: the Windows Crypto Application Programmer’s Interface (API) library and OpenSSL. The entropy source used by both is the underlying Windows Server 2019 platform, which is considered to be in the TOE Operational Environment. Since the TOE vendor does not have access to design information or raw entropy data, detailed entropy source description cannot be provided.

Design description

[EAR] section 2, "Entropy Source Design," provides as much description of the design of the entropy source used by the TOE as is possible. Academic papers and FIPS validations are cited to provide some conventional wisdom about the implementation of the platform entropy source.

Operation of the entropy source

[EAR] section 2, "Entropy Source Design," provides some general information about the operation of the entropy source, but since it is a third-party implementation, the TOE vendor cannot describe it in detail. This section provides information about how the TOE uses entropy gathered from the platform to generate its random data using both the Windows crypto API and the OpenSSL API.

Entropy justification

[EAR] section 2, "Entropy Source Design," cannot provide justification for the amount of entropy generated by the platform, but does provide the amount of entropy required by the TOE, 256 bits of entropy is assumed to be present, and why that amount is expected to be available.

Section 2.1 describes the Windows Entropy Source and provides rationale for the assumption it provides 256 bits of entropy to initialize the SP800-90A DRBG based on FIPS 140-2 validation of the underlying OS.

Section 2.2 describes the OpenSSL DRBG seeding process on Windows and provides rationale for the assumption it provides 256 bits of entropy based on it being seeded with 384 bits of entropy from the entropy pool as specified by SP800-90A Table 3.

Operating conditions

[EAR] section 3, "Operating Conditions," summarizes the operating conditions required by the entropy source used by the TOE. It states Windows Server 2019 is meant to operate in a server room at regular server room temperature and humidity levels.

Health testing

[EAR] section 4, "Health Testing," states that the OpenSSL DRBG implements a continuous test for RNGs as defined by FIPS 140-2, but any self-test performed by the Windows platform is unknown due to the proprietary nature of the platform.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_RBG_EXT.2.2-ATE-01

> *In the future, specific statistical testing (in line with NIST SP 800-90B) will be required to verify the entropy estimates.*

Summary

This text in [ASPPv1.3] is informational only, no test is specified. The evaluator determines it is implicitly satisfied.

2.3.1.11 Storage of Credentials (FCS_STO_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FCS_STO_EXT.1-ASE-01

> *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.*

Summary

Section 7.1.1.3, "Storage of credentials," lists the credentials required by the TOE, the purpose for the credential, and the manner it is stored.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_STO_EXT.1-ATE-01

For all credentials for which the application implements functionality, the evaluator shall verify credentials are encrypted according to FCS_COP.1(1) or conditioned according to FCS_CKM.1.1(1) and FCS_CKM.1(3).

For all credentials for which the application invokes platform-provided functionality, the evaluator shall perform the following actions which vary per platform.

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

For 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.*

For iOS:*The evaluator shall verify that all credentials are stored within a Keychain.*

For Linux: *The evaluator shall verify that all keys are stored using Linux keyrings.*

For Solaris: *The evaluator shall verify that all keys are stored using Solaris Key Management Framework (KMF).*

For macOS:*The evaluator shall verify that all credentials are stored within Keychain.*

Summary

The evaluator installed the TOE with EFS and verified all files created by the TOE are not accessible by other users or non-administrators. The evaluator also verified that all TOE certificates are present in the Windows Certificate Store and all sensitive registry entries use DPAPI.

2.3.1.12 TLS Protocol (FCS_TLS_EXT.1)

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FCS_TLS_EXT.1-AGD-01

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

Summary

Work unit ASE_REQ.1-10 states that the evaluator examined the security requirements for consistency. The requirements are specified unambiguously and do not contradict themselves or other parts of the [ST]. More specifically, per the [ST], section 6.1.1.12, *Protocol (FCS_TLS_EXT.1)*, indicates the selection "**TLS as a client**." This is the only SFR in [TLSPKGv1.1] that determines further selections in dependent components. The evaluator examined the [ST] and determined that the correct dependent components were selected, specifically FCS_TLSC_EXT.1 and FCS_TLSC_EXT.5.

Test Assurance Activities

No assurance activities defined.

2.3.1.13 TLS Client Protocol (FCS_TLSC_EXT.1)

FCS_TLSC_EXT.1.1

TSS Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.1-ASE-01

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.

Summary

[ST] section 7.1.1.4, "HTTPS and TLS Protocols," describes the implementation of TLS. The TLS ciphersuite listed, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289, is the same one listed in FCS_TLSC_EXT.1 in chapter 6.

Guidance Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.1-AGD-01

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.

Summary

The evaluator examined the [ST], in particular Section 7.1.1.4, *HTTPS and TLS Protocols* (part of the TSS) and found the following description of the use of TLS.

- TLS version 1.2 must be used.
- The TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 cypher suite must be used.
- TLS certificates will be automatically created.

The evaluator reviewed section 2 of [CC-CFG] and verified that it describes how the user shall configure the TLS for the TOE as follows.

- Section 2.1 specifies how to make TLS 1.2 the default TLS setting on Windows Server 2019.
- Section 2.2 specifies how to enable FIPS and NIAP modes for the TOE, which limits TLS to v1.2, limits the ciphers, and disables modules' updates.
- Section 2.5 specifies that it is not necessary for the administrator to generate and install TLS server certificates since the installation and configuration process will automatically create them.

The evaluator determined that the guidance contains instructions on configuring TLS with the TOE to conform to the description in the TSS.

Test Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.1-ATE-01

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).*
- ***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.
- ***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.*
- ***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.*
- ***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.*
  - ***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.*
  - ***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.*
  - ***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.*
  - ***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.*
  - ***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.*
  - ***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.*

Summary

Test 1: The evaluator successfully established connections for the ciphersuite implemented by the TOE Platform as well as the TOE implemented TLSC client (OpenSSL), both using TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256.

Test 2: The evaluator was able to establish connections to the OpenSSL proxy server using certificates with ServerAuth present as an extended key usage field. When the ServerAuth EKU was removed, the TLS clients would not establish a connection.

Test 3: The evaluator was unable to establish connections when the selected ciphersuite did not match the server certificate.

Test 4: The evaluator was unable to establish connections when presented with the TLS_NULL_WITH_NULL_NULL cipher suite.

Test 5.1: The evaluator was unable to establish connections when presented with TLS version 1.4.

Test 5.2: The evaluator was unable to establish connections when presented with TLS version 1.3, which is not supported in the evaluated configuration.

Test 5.3: The evaluator was unable to establish connections when the Server Hello nonce was modified.

Test 5.4: The evaluator was unable to establish connections when the Server Hello ciphersuite was modified.

Test 5.5: The evaluator was unable to establish connections when the Server Hello signature block in the Key Exchange handshake message was modified.

Test 5.6: The evaluator was unable to establish connections when the Server Finished handshake message was modified.

Test 5.7: The evaluator was unable to establish connections when a garbled message is received from the Server after the Server has issued the ChangeCipherSpec message.

FCS_TLSC_EXT.1.2

TSS Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.2-ASE-01

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.

Summary

[ST] section 7.1.1.4, "HTTPS and TLS Protocols," describes how the TOE handles reference identifiers and states that CN and SAN are the only supported reference identifiers. The section also states wildcards are accepted only as the leftmost portion of the reference identifier and that certificate pinning is not supported.

Guidance Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.2-AGD-01

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.

Summary

The evaluator examined the [ST] section 7.1.1.4, *HTTPS and TLS Protocols*, and found that the common name (CN) and subject alternate name (SAN) are the only supported reference identifiers that can be forced as part of the certificate validation, and this behavior is not configurable. Therefore, guidance on how to set the reference identifier would be inaccurate. However, Section 2.5, *TLS Server Certificates*, of [CC-CFG] specifies that it is not necessary for the administrator to generate and install TLS server certificates since the installation and configuration process will automatically create them.

Test Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.2-ATE-01

[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.
- ***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.*
- ***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.*
- ***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.*
- ***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 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.*
  - ***Test 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.*
  - ***Test 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.*
  - ***Test 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 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.*
- ***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.*

Summary

Test 1: The evaluator was unable to establish connections with a server presenting a certificate containing a CN or SAN extension not matching the reference identifier.

Test 2: The evaluator was unable to establish connections with the test server presenting a certificate with a CN that matches the reference identifier but a SAN extension that does not match the reference identifier.

Test 3: The evaluator was able to establish connections with the test server when the CN matches the reference identifier but does not contain a SAN.

Test 4: The evaluator was able to establish connections with the test server when the CN does not match the reference identifier but the SAN extension does.

Test 5.1: The evaluator was unable to establish connections with the test server when the wildcard is not in the left-most label of the presented identifier.

Test 5.2: The evaluator was able to establish connections with the test server when a valid wildcard is presented along with a matching reference identifier but was unable to establish connection when these are applied incorrectly.

Test 5.3: The evaluator was unable to establish connections with the test server using the criteria identified in the assurance activity.

Test 5.4: Per the [ST], the TOE does support wildcards in certificates, therefore this test is not applicable.

Test 6: Per the [ST], the TOE does not support URI or Service name reference in the SAN, therefore this test is not applicable.

Test 7: Per the [ST], the TOE does not support pinned certificates, therefore this test is not applicable.

FCS_TLSC_EXT.1.3

TSS Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.3-ASE-01

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.

Summary

The authorized override exception is not selected in FCS_TLSC_EXT.1.3, therefore the evaluator determines this work unit is not applicable to the TOE.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.1.3-ATE-01

[TD-0513] The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted:

- ***Test 1a:** The evaluator shall demonstrate that a server using a certifcate 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.*
- ***Test 2:** The evaluator shall demonstrate that a server using a certificate which has been revoked results in an authentication failure.*

- ***Test 3:*** *The evaluator shall demonstrate that a server using a certificate which has passed its expiration date results in an authentication failure.*
- ***Test 4:*** *The evaluator shall demonstrate that a server using a certificate which does not have a valid identifier results in an authentication failure.*

Summary

Test 1a: The evaluator was able to establish connections to a server using a certificate with valid certification path.

Test 1b: The evaluator was unable to establish connections to a server using a certificate without a valid certification path.

Test 1c: The TOE does not manage a trust store therefore this test is not applicable.

Test 2: The evaluator was unable to establish connections to a server using a revoked certificate.

Test 3: The evaluator was unable to establish connections to a server using an expired certificate.

Test 4: This test is performed in conjunction with the test for FCS_TLSC_EXT.1.2 Test 1.

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

TSS Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.5-ASE-01

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

Summary

[ST] section 7.1.1.4, "HTTPS and TLS Protocols," describes the TOE support of the Support Groups Extension.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FCS_TLSC_EXT.5-ATE-01

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.*

Summary

The evaluator was able to establish connections to the server using each of the TOE's supported curves ECDHE (P-256, P-384).

2.3.2 User data protection (FDP)

2.3.2.1 Encryption Of Sensitive Application Data (FDP_DAR_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FDP_DAR_EXT.1-ASE-01

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.*

Summary

[ST] section 7.1.2.1, "Encryption of Sensitive Application Data," describes what sensitive information is stored by the TOE and that it is stored in the platform-provided encrypted file system.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDP_DAR_EXT.1-ATE-01

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 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.*

***For Android:** The evaluator shall inspect the TSS and verify that it describes how files containing sensitive data are stored with the MODE_PRIVATE flag set.*

***For Windows:**The Windows platform currently does not provide data-at-rest encryption services which depend upon invocation by application developers. The evaluator shall verify that the Operational User Guidance makes the need to activate platform encryption, such as BitLocker or Encrypting File System (EFS), clear to the end user.*

***For iOS:**The evaluator shall inspect the TSS and ensure that it describes how the application uses the Complete Protection, Protected Unless Open, or Protected Until First User Authentication Data Protection Class for each data file stored locally.*

***For Linux:** The Linux platform currently does not provide data-at-rest encryption services which depend upon invocation by application developers. The evaluator shall verify that the Operational User Guidance makes the need to activate platform encryption clear to the end user.*

***For Solaris:** The Solaris platform currently does not provide data-at-rest encryption services which depend upon invocation by application developers. The evaluator shall verify that the Operational User Guidance makes the need to activate platform encryption clear to the end user.*

***For macOS:** The macOS platform currently does not provide data-at-rest encryption services which depend upon invocation by application developers. The evaluator shall verify that the Operational User Guidance makes the need to activate platform encryption clear to the end user.*

Summary

The evaluator verified that the user guidance [CC-CFG] clearly described the requirement and procedure for enabling Encrypting File System (EFS). Therefore all data written by the TOE is encrypted using platform-provided functionality.

2.3.2.2 Access to Platform Resources (FDP_DEC_EXT.1)

FDP_DEC_EXT.1.1

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FDP_DEC_EXT.1.1-AGD-01

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.

Summary

The evaluator examined user documentation, both [CC-CFG] and [ADM_GUIDE]. Per [ADM_GUIDE] section 2, *About the Cloud Extender*, the base functionality of the TOE is to integrate on-premise systems such as email, directories, certificate authorities, and application and content servers. Further, section 4, *Cloud Extender Architecture*, indicates that the TOE is installed "behind the firewall with network access to the appropriate internal systems." Therefore, network hardware access is inherent to the functionality of the TOE.

The only selection available for hardware resources in the [ASPPv1.3] is in *FDP_NET_EXT.1 Network communications*. Per the [ST] section 6.1.2.2., *Network Communications (FDP_NET_EXT.1)*, the application restricts network communication to a list of network communications. This is consistent with the documentation as indicated in the previous paragraph.

Test Assurance Activities

Assurance Activity AA-FDP_DEC_EXT.1.1-ATE-01

For Android: *[TD0515] 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.*

For Windows: *[TD0434] 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.

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

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

For Solaris: *The evaluator shall verify that either the application software or its documentation provides a list of the hardware resources it accesses.*

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

Summary

The TOE is a Windows Server service, neither a Windows Universal Application nor a Windows Desktop Application, though the latter is closer.

The TOE requires access to no sensitive information repositories.

FDP_DEC_EXT.1.2

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FDP_DEC_EXT.1.2-AGD-01

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.

Summary

The evaluator examined user documentation, both [CC-CFG] and [ADM_GUIDE]. Per [ADM_GUIDE] section 2, *About the Cloud Extender*, the base functionality of the TOE is to integrate on-premise systems such as email, directories, certificate authorities, and application and content servers. Therefore, access to these is inherent to the functionality of the TOE.

The relevant selection in the [ASPPv1.3] is *FDP_DEC_EXT.1.2*. Per the [ST] section 6.1.2.1., *Access to Platform Resources (FDP_DEC_EXT.1)*, the application restricts its access to the following.

- No sensitive information repositories
- Address book
- Calendar
- Call lists
- System logs
- The Windows Credential Store
- System logs and Windows Event logs-application with the following folders and sub-folders
  - C:\Program Files (x86)\MaaS360\Cloud Extender
  - C:\ProgramData\MaaS360\Cloud Extender
  - C:\Program Files (x86)\Common Files\MaaS360\Visibility_2.106.500.016.002

This is consistent with the documentation as indicated in the previous paragraph.

Test Assurance Activities

Assurance Activity AA-FDP_DEC_EXT.1.2-ATE-01

For Android: *[TD0515] The evaluator shall verify that each <uses-permission> entry in the AndroidManifest.xml file for access to a sentitive information repository is reflected in the selection.*

For 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.

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

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

For Solaris:*The evaluator shall verify that either the application software or its documentation provides a list of sensitive information repositories it accesses.*

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

Summary

The TOE is a Windows Server service, neither a Windows Universal Application nor a Windows Desktop Application, though the latter is closer.

The TOE requires access to no sensitive information repositories.

2.3.2.3 Network Communications (FDP_NET_EXT.1)

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FDP_NET_EXT.1-ATE-01

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.*
- ***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).*

For Android: *If "no network communication" is selected, the evaluator shall ensure that the application's AndroidManifest.xml file does not contain a uses-permission or uses-permission-sdk-23 tag containing android:name="android.permission.INTERNET". In this case, it is not necessary to perform the above Tests 1 and 2, as the platform will not allow the application to perform any network communication.*

Summary

The evaluator inspected the network traffic (test 1) and scanned network ports (test 2, during AVA) while the application was running. The evaluator was able to verify that all ports and network traffic are defined in the TSS.

2.3.3 Identification and authentication (FIA)

2.3.3.1 X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1

TSS Assurance Activities

Assurance Activity AA-FIA_X509_EXT.1.1-ASE-01

[TD0601] 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.

Summary

[ST] section 7.1.3.1, "X.509 Certificate Validation," explains how the TOE validates certificates. It is stated that the TOE uses X.509 in support of TLS authentication, thus certificate validity checking is performed for TLS connections, including HTTPS. It is also stated certificate path validation conforms to RFC 5280.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_X509_EXT.1.1-ATE-01

[TD0601] 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.
- ***Test 2:*** *The evaluator shall demonstrate that validating an expired certificate results in the function failing.*
- ***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 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.

- ***Test 4:** If any OCSP option is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails.*
- ***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.)*
- ***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.)*
- ***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.)*
- ***Test 8a:** (Conditional on support for EC certificates as indicated in FCS_COP.1(3)). 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.*
- ***Test 8b:** (Conditional on support for EC certificates as indicated in FCS_COP.1(3)). 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.*

Summary

The evaluator constructed a test environment with a root certificate and two certificate authorities with CRL signing and basicConstraints extensions. The evaluator also created another certificate authority without these capabilities. Certificates were then generated in order to facilitate the testing requirements.

Test 1: The evaluator was able to establish connections when a certificate chain could be validated and unable to establish connections when a certificate chain could not be validated.

Test 2: The evaluator was unable to establish connections when presented with an expired certificate.

Test 3: The evaluator was unable to establish connections when presented with a revoked certificate after having successfully established a connection with a valid certificate.

Test 4: The evaluator was unable to establish connections when presented with a certificate from the CA that did not have the CRLsign key usage.

Test 5: The evaluator was unable to establish connections when presented with a certificate that had been modified in the first eight bytes.

Test 6: The evaluator was unable to establish connections when presented with a certificate that had been modified in the last eight bytes.

Test 7: The evaluator was unable to establish connections when presented with a certificate that had been modified in any byte of the public key.

Test8a is not applicable as the [ST] does not claim EC certificates.

Test8b is not applicable as the [ST] does not claim EC certificates.

FIA_X509_EXT.1.2

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_X509_EXT.1.2-ATE-01

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:*** *[TD0495] 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.*
- ***Test 2:*** *[TD0495] 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.*

Summary

The evaluator constructed a test environment with a root certificate and two certificate authorities with CRL signing and basicConstraints extensions. The evaluator created another certificate authority without these capabilities. Certificates were then generated in order to facilitate the testing requirements.

Test 1: The evaluator could not establish a connection when the basicConstraints extension was not present for the intermediate CA.

Test 2: The evaluator could not establish a connection when the basicConstraints extension was not set for the intermediate CA.

2.3.3.2 X.509 Certificate Authentication (FIA_X509_EXT.2)

TSS Assurance Activities

Assurance Activity AA-FIA_X509_EXT.2-ASE-01

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.

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. 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.

Summary

[ST] section 7.1.3.1, "X.509 Certificate Validation," describes the certification validation process, including the SAN/CN check used to determine an applicable certificate. Certificates for the platform are stored in the Windows Credential Store.

This section also states that the TOE does not support addition or configuration of additional TLS certificates and that invalid certificates or certificates where revocation status cannot be determined are rejected.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FIA_X509_EXT.2-ATE-01

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.*

Summary

Test 1 proper connection was demonstrated during FIA_X509_EXT.1.1 Test 1. For the negative result, the evaluator caused a CRL failure by shutting down the CRL server and attempting to make the connection. The connection was not established.

For Test 2, the evaluator performed this test in conjunction with FIA_X509_EXT.1.1 Test 3.

2.3.4 Security management (FMT)

2.3.4.1 Secure by Default Configuration (FMT_CFG_EXT.1)

FMT_CFG_EXT.1.1

TSS Assurance Activities

Assurance Activity AA-FMT_CFG_EXT.1.1-ASE-01

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

Summary

[ST] section 7.1.4.1, "Secure By Default Configuration," states that the TOE does not require any credentials for access control to the application and provides a list of default certificates which are installed with the TOE.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FMT_CFG_EXT.1.1-ATE-01

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.*

Summary

The evaluator verified that during installation, the only application functionality available was to set new credentials (test 2). The only credential is the TOE license key and the evaluator only has one, so it can only be removed by reinstalling the TOE, not replaced with a different one, therefore test 3 is not applicable.

FMT_CFG_EXT.1.2

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FMT_CFG_EXT.1.2-ATE-01

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.

For Android:*The evaluator shall run ls -alR|grep -E '^.....w.' inside the application's data directories to ensure that all files are not world-writable. The command should not print any files. The evaluator shall also verify that no sensitive data is written to external storage which could be read/modified by any application containing the READ_EXTERNAL_STORAGE and/or WRITE_EXTERNAL_STORAGE permissions.*

For 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.*

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

For 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.

For Solaris:*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.*

For 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.*

Summary

Using icacls.exe the evaluator verified that files written to disk during an applications installation have the correct file permissions, such that a standard user cannot modify the application or its data files.

2.3.4.2 Supported Configuration Mechanism (FMT_MEC_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FMT_MEC_EXT.1-ASE-01

[TD0437] 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.

[TD0437] 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.

Summary

[ST] section 7.1.4.2, "Supported Configuration Mechanism," states that the configuration data is stored in the platform's Windows Registry or may be optionally exported to a file in a platform-encrypted folder. The configuration options are listed in Table 19, including those specified by the operational guidance ([CC-CFG]) to mandate the use of TLS 1.2 and EFS.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FMT_MEC_EXT.1-ATE-01

[TD0437] 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:

For 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 at location /data/data/package/shared_prefs/ reflects the changes made to the configuration to verify that the application used SharedPreferences and/or PreferenceActivity classes for storing configuration data, where package is the Java package of the application.*

For 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.

For iOS:*The evaluator shall verify that the app uses the user defaults system or key-value store for storing all settings.*

For 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).*

For Solaris:*The evaluator shall run the application while monitoring it with the utility dtrace. The evaluator shall make security-related changes to its configuration. The evaluator shall verify that dtrace logs corresponding changes to configuration files that reside in /etc (for system-specific configuration) or in the user's home directory(for user-specific configuration).*

For 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.

Summary

The TOE is a Windows Server service, neither a Windows Universal Application nor a Windows Desktop Application, though the latter is closer.

With ProcMon started and the TOE running, the evaluator made configuration changes to all four modules in scope and observed that the files modified are part of the TOE. He also confirmed registry changes were made only in appropriate locations and that DPAPI was used to protect the information.

2.3.4.3 Specification of Management Functions (FMT_SMF.1)

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FMT_SMF.1-AGD-01

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.

Summary

FMT_SMF.1.1 in the [ST] specifies that "The TSF shall be capable of performing the following management functions: **configure cryptographic functionality.**" has been selected by the ST author. The evaluator reviewed section 2 of [CC-CFG] and verified that it describes how the user shall configure the cryptographic functionalities for the TOE as follows.

- Section 2.1 specifies how to make TLS 1.2 the default TLS setting on the Windows Server 2019.
- Section 2.2 specifies how to enable FIPS and NIAP modes for the TOE, which limits TLS to v1.2, limits the ciphers, and disables modules' updates.
- Section 2.3 specifies how to create an Exchange Server certificate.
- Section 2.4 specifies how to enable WinRM for HTTPS.

- Section 2.5 specifies that it is not necessary for the administrator to generate and install TLS server certificates since the installation and configuration process will automatically create them.

After reviewing the evidence, the evaluator verified that every management function claimed by the [ST] is described in the operational environment guidance.

Test Assurance Activities

Assurance Activity AA-FMT_SMF.1-ATE-01

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.

Summary

The selection claimed in [ST] is a one-time configuration of platform cryptographic functionality performed at TOE installation as specified in section 2.1 of [CC-CFG]. That it achieves the proper configuration is tested by the platform-related TLS tests.

2.3.5 Privacy (FPR)

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

TSS Assurance Activities

Assurance Activity AA-FPR_ANO_EXT.1-ASE-01

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

Summary

[ST] section 7.1.5.1, "User Consent for Transmission of PII," states that the TOE does not contain any functionality relating to Personally Identifiable Information.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPR_ANO_EXT.1-ATE-01

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

Summary

Section 7.1.5.1 of [ST] states the application contains no functionality relating to Personally Identifiable Information (PII). The evaluator determined that since no PII is transmitted by the TOE, this test is not applicable.

2.3.6 Protection of the TSF (FPT)

2.3.6.1 Anti-Exploitation Capabilities (FPT_AEX_EXT.1)

FPT_AEX_EXT.1.1

TSS Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.1.1-ASE-01

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

Summary

[ST] section 7.1.6.1, "Anti-exploitation Capabilities," states the TOE is built using the /DYNAMICBASE and /GS compiler flags and linking option to enable ASLR.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.1.1-ATE-01

[TD0544] 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.

***For Android:** [TD0544] 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.*

***For 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.*

***For 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.*

***For 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.*

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

***For macOS:**The evaluator shall run the same application on two different Mac systems. The evaluator shall then compare their memory maps using vmmap PID to ensure the two different instances share no mapping locations.*

Summary

The evaluator used the VMMap tool to dump the memory of binary executable component of the Cloud Extender. Three different memory dumps on three different machines were made and compared and the evaluator determined no memory location was consistently repeated.

The evaluator also ran BinScope against the TOE component and verified the ASLR checks passed.

FPT_AEX_EXT.1.2

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.1.2-ATE-01

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

For Android: *The evaluator shall perform static analysis on the application to verify that*

- *mmap is never invoked with both the PROT_WRITE and PROT_EXEC permissions, and*
- *mprotect is never invoked.*

For 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.*

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

For 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.*

For Solaris: *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.*

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

Summary

The evaluator has confirmed the TOE binary pass the /NXCOMPACT check using the BinScope utility and that DEP protection is enabled.

FPT_AEX_EXT.1.3

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.1.3-ATE-01

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

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

For 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).*

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

For Linux: *[TD0435] The evaluator shall ensure that the application can successfully run on a system with either SELinux or AppArmor enabled and in enforce mode.*

For Solaris:*The evaluator shall ensure that the application can run with Solaris Trusted Extensions enabled and enforcing.*

For macOS: *The evaluator shall ensure that the application can successfully run on macOS without disabling any security features.*

Summary

The evaluator used the Windows Security GUI to configure the Windows Defender Exploit Guard functionality of Windows Server 2019 to protect the TOE binary and was then able to run the TOE successfully with the protection enabled.

FPT_AEX_EXT.1.4

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.1.4-ATE-01

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:

For 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.*

For Windows: *[TD0445] 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.*

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

For 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.*

For Solaris: *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.*

For 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.*

Summary

The TOE is a Windows Server service, neither a Windows Universal Application nor a Windows Desktop Application, though the latter is closer.

The evaluator examined the application directories and confirmed that there were no executable files stored in the same directories to which the application wrote data.

FPT_AEX_EXT.1.5

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_AEX_EXT.1.5-ATE-01

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

For 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.

Summary

The evaluator used the BinScope utility and verified that all TOE native binaries pass the /GS check and that the use of the /GS flag was present in the TSS.

The evaluator used WinDbgx64 to find the expected code sequences in PE native executables.

2.3.6.2 Use of Supported Services and APIs (FPT_API_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FPT_API_EXT.1-ASE-01

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

Summary

[ST] section 7.1.6.2, "Use of Supported Services and APIs," lists the APIs provided by the platform used by the TOE.

Table 2: Windows APIs Used by the Cloud Extender

API Category	Windows APIs
PowerShell Commandlets	• Get-ItemProperty https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-itemproperty • Get-Item https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-item • Remove-Item https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/remove-item • Get-Command https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/get-command • Add-Type https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/add-type • New-Object https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/new-object • Get-PSSession https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/get-pssession • Remove-PSSession https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/remove-pssession • ConvertTo-SecureString https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.security/convertto-securestring • New-PSSessionOption https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/new-pssessionoption • Connect-ExchangeOnline https://docs.microsoft.com/en-us/powershell/module/exchange/connect-exchangeonline • Start-Sleep https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/start-sleep • Import-PSSession

API Category	Windows APIs
	https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/import-pssession • Get-Content https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-content • get-childitem https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-childitem • Export-CSV https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/export-csv • Get-PSSnapin https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/get-pssnapin • Add-PSSnapin https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/add-pssnapin • Remove-Variable https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/remove-variable • Get-ActiveSyncOrganizationSettings https://docs.microsoft.com/en-us/powershell/module/exchange/get-activesyncorganizationsettings • Set-ActiveSyncOrganizationSettings https://docs.microsoft.com/en-us/powershell/module/exchange/set-activesyncorganizationsettings • Get-ExchangeServer https://docs.microsoft.com/en-us/powershell/module/exchange/get-exchangeserver • Get-ActiveSyncMailboxPolicy https://docs.microsoft.com/en-us/powershell/module/exchange/get-activesyncmailboxpolicy • Get-MobileDeviceMailboxPolicy https://docs.microsoft.com/en-us/powershell/module/exchange/get-mobiledevicemailboxpolicy • Set-ActiveSyncMailboxPolicy https://docs.microsoft.com/en-us/powershell/module/exchange/set-activesyncmailboxpolicy • Set-MobileDeviceMailboxPolicy https://docs.microsoft.com/en-us/powershell/module/exchange/set-mobiledevicemailboxpolicy • New-ActiveSyncMailboxPolicy https://docs.microsoft.com/en-us/powershell/module/exchange/new-activesyncmailboxpolicy • New-MobileDeviceMailboxPolicy https://docs.microsoft.com/en-us/powershell/module/exchange/new-mobiledevicemailboxpolicy • Remove-ActiveSyncMailboxPolicy https://docs.microsoft.com/en-us/powershell/module/exchange/remove-activesyncmailboxpolicy • Remove-MobileDeviceMailboxPolicy https://docs.microsoft.com/en-us/powershell/module/exchange/remove-mobiledevicemailboxpolicy

API Category	Windows APIs
	• Get-CASMailbox https://docs.microsoft.com/en-us/powershell/module/exchange/get-casmailbox • Set-CASMailbox https://docs.microsoft.com/en-us/powershell/module/exchange/set-casmailbox • Get-ActiveSyncDeviceStatistics https://docs.microsoft.com/en-us/powershell/module/exchange/get-activesyncdevicestatistics • Get-ActiveSyncDevice https://docs.microsoft.com/en-us/powershell/module/exchange/get-activesyncdevice • Get-MobileDeviceStatistics https://docs.microsoft.com/en-us/powershell/module/exchange/get-mobiledevicestatistics • Get-MobileDevice https://docs.microsoft.com/en-us/powershell/module/exchange/get-mobiledevice • Clear-ActiveSyncDevice https://docs.microsoft.com/en-us/powershell/module/exchange/clear-activesyncdevice • Clear-MobileDevice https://docs.microsoft.com/en-us/powershell/module/exchange/clear-mobiledevice • Remove-ActiveSyncDevice https://docs.microsoft.com/en-us/powershell/module/exchange/remove-activesyncdevice • Remove-MobileDevice https://docs.microsoft.com/en-us/powershell/module/exchange/remove-mobiledevice • Get-OrganizationalUnit https://docs.microsoft.com/en-us/powershell/module/exchange/get-organizationalunit • Get-Recipient https://docs.microsoft.com/en-us/powershell/module/exchange/get-recipient • Remove-RoleGroup https://docs.microsoft.com/en-us/powershell/module/exchange/remove-rolegroup • Remove-ManagementRoleAssignment https://docs.microsoft.com/en-us/powershell/module/exchange/remove-managementroleassignment • Remove-ManagementRole https://docs.microsoft.com/en-us/powershell/module/exchange/remove-managementrole • New-ManagementRole https://docs.microsoft.com/en-us/powershell/module/exchange/new-managementrole • Get-ManagementRoleEntry https://docs.microsoft.com/en-us/powershell/module/exchange/get-managementroleentry • Read-Host

API Category	Windows APIs
	https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/read-host • Write-Host https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/write-host • New-RoleGroup https://docs.microsoft.com/en-us/powershell/module/exchange/new-rolegroup
Active Directory	• Forest.GetCurrentForest https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.activedirectory.forest.getcurrentforest • DirectoryEntry.Properties https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.directoryentry.properties • DirectorySearcher.FindAll https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.directorysearcher.findall • DirectorySearcher.PropertiesToLoad.Add https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.directorysearcher.propertiestoload • System.DirectoryServices.ActiveDirectory.Forest.GetCurrentForest().GetAllTrustRelationships https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.activedirectory.forest.getalltrustrelationships • Domain.GetDomain https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.activedirectory.domain.getdomain
LDAP	• SearchRequest https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.protocols.searchrequest • DirectoryAttributeModification https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.protocols.directoryattributemodification • ModifyRequest https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.protocols.modifyrequest • LdapConnection https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.protocols.ldapconnection • VerifyServerCertificateCallback https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.protocols.verifyservercertificatecallback • DirectoryAttribute https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.protocols.directoryattribute • PageResultRequestControl https://docs.microsoft.com/en-us/dotnet/api/system.directoryservices.protocols.pageresultrequestcontrol
Windows Registry	• RegOpenCurrentUser https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regopencurrentuser • RegCreateKey

API Category	Windows APIs
	https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regcreatekeya • RegOpenKey https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regopenkeya • RegCloseKey https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regclosekey • RegDeleteKey https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regdeletekeya • RegDeleteValue https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regdeletevaluea • RegQueryInfoKey https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regqueryinfokeya • RegQueryValue https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regqueryvaluea • RegSetValue https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regsetvaluea • RegEnumValue https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regenumvaluea • RegEnumKey https://docs.microsoft.com/en-us/windows/win32/api/winreg/nf-winreg-regenumkeya
Windows Management Instrumentation (WMI)	• ExecQuery https://docs.microsoft.com/en-us/windows/win32/api/wbemcli/nf-wbemcli-iwbemservices-execquery • CoInitializeEx https://docs.microsoft.com/en-us/windows/win32/api/combaseapi/nf-combaseapi-coinitializeex • CoUninitialize https://docs.microsoft.com/en-us/windows/win32/api/combaseapi/nf-combaseapi-couninitialize • GetObjectText https://docs.microsoft.com/en-us/windows/win32/api/wbemcli/nf-wbemcli-iwbemclassobject-getobjecttext • ExecMethod https://docs.microsoft.com/en-us/windows/win32/api/wbemcli/nf-wbemcli-iwbemservices-execmethod • SpawnInstance https://docs.microsoft.com/en-us/windows/win32/api/wbemcli/nf-wbemcli-iwbemclassobject-spawninstance • ConnectServer https://docs.microsoft.com/en-us/windows/win32/api/wbemcli/nf-wbemcli-iwbemlocator-connectserver • CreateObjectStub https://docs.microsoft.com/en-us/windows/win32/api/wbemcli/nf-wbemcli-iunsecuredapartment-createobjectstub

API Category	Windows APIs
	• ExecNotificationQueryAsync https://docs.microsoft.com/en-us/windows/win32/api/wbemcli/nf-wbemcli-iwbemservices-execnotificationqueryasync • CancelAsyncCall https://docs.microsoft.com/en-us/windows/win32/api/wbemcli/nf-wbemcli-iwbemservices-cancelasynccall • GetObjectW https://docs.microsoft.com/en-us/windows/win32/api/wingdi/nf-wingdi-getobjectw
Process	• CreateProcess https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-createprocessa • GetProcAddress https://docs.microsoft.com/en-us/windows/win32/api/libloaderapi/nf-libloaderapi-getprocaddress • GetExitCodeProcess https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-getexitcodeprocess • CloseHandle https://docs.microsoft.com/en-us/windows/win32/api/handleapi/nf-handleapi-closehandle • AdjustTokenPrivileges https://docs.microsoft.com/en-us/windows/win32/api/securitybaseapi/nf-securitybaseapi-adjusttokenprivileges • LookupPrivilegeValue https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-lookupprivilegevaluea • OpenProcessToken https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-openprocesstoken • OpenProcess https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-openprocess • OpenThreadToken https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-openthreadtoken • CreateToolhelp32Snapshot https://docs.microsoft.com/en-us/windows/win32/api/tlhelp32/nf-tlhelp32-createtoolhelp32snapshot • Process32First https://docs.microsoft.com/en-us/windows/win32/api/tlhelp32/nf-tlhelp32-process32first • Process32Next https://docs.microsoft.com/en-us/windows/win32/api/tlhelp32/nf-tlhelp32-process32next • TerminateProcess https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-terminateprocess • WaitForMultipleObjects https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-waitformultipleobjects • WaitForSingleObject

API Category	Windows APIs
	https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-waitforsingleobject • CreateMutex https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-createmutexa • OpenMutex https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-openmutexw • Sleep https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-sleep • ReleaseMutex https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-releasemutex • ReleaseSemaphore https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-releasesemaphore • CreateSemaphore https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-createsemaphorea • OpenSemaphore https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-opensemaphorew • CreateEvent https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-createeventa • ResetEvent https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-resetevent • GetCurrentProcess https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-getcurrentprocess • GetProcessIoCounters https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-getprocessiocounters • GetProcessMemoryInfo https://docs.microsoft.com/en-us/windows/win32/api/psapi/nf-psapi-getprocessmemoryinfo • SetThreadPriority https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-setthreadpriority • SetEvent https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-setevent • GetCurrentThreadId https://docs.microsoft.com/en-us/windows/win32/api/processthreadsapi/nf-processthreadsapi-getcurrentthreadid • RegisterEventSource https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-registereventsourcea • ReportEvent https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-reporteventa

API Category	Windows APIs
	• DeregisterEventSource https://docs.microsoft.com/en-us/windows/win32/api/winbase/nf-winbase-deregistereventsource • PeekMessage https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-peekmessagea • TranslateMessage https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-translatemessage • DispatchMessage https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-dispatchmessage • EnterCriticalSection https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-entercriticalsection • LeaveCriticalSection https://docs.microsoft.com/en-us/windows/win32/api/synchapi/nf-synchapi-leavecriticalsection • IsWow64Process https://docs.microsoft.com/en-us/windows/win32/api/wow64apiset/nf-wow64apiset-iswow64process • CoCreateInstance https://docs.microsoft.com/en-us/windows/win32/api/combaseapi/nf-combaseapi-cocreateinstance
Windows OS HTTP (Networking)	• WinHttpSendRequest https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpsendrequest • WinHttpWriteData https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpwritedata • WinHttpReceiveResponse https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpreceiveresponse • WinHttpQueryDataAvailable https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpquerydataavailable • WinHttpReadData https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpreaddata • WinHttpOpen https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpopen • WinHttpSetTimeouts https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpsettimeouts • WinHttpCrackUrl https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpcrackurl • WinHttpConnect https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpconnect • WinHttpOpenRequest

API Category	Windows APIs
	https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpopenrequest • WinHttpCloseHandle https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpclosehandle • WinHttpSetOption https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpsetoption • WinHttpSetCredentials https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpsetcredentials • WinHttpAddRequestHeaders https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpaddrequestheaders • WinHttpQueryHeaders https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpqueryheaders • WinHttpQueryAuthSchemes https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpqueryauthschemes • WSACloseEvent https://docs.microsoft.com/en-us/windows/win32/api/winsock2/nf-winsock2-wsacloseevent • WSAGetLastError https://docs.microsoft.com/en-us/windows/win32/api/winsock/nf-winsock-wsagetlasterror • WSACreateEvent https://docs.microsoft.com/en-us/windows/win32/api/winsock2/nf-winsock2-wsacreateevent • NotifyAddrChange https://docs.microsoft.com/en-us/windows/win32/api/iphlpapi/nf-iphlpapi-notifyaddrchange • CoSetProxyBlanket https://docs.microsoft.com/en-us/windows/win32/api/combaseapi/nf-combaseapi-cosetproxyblanket • WinHttpGetIEProxyConfigForCurrentUser https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpgetieproxyconfigforcurrentuser • InternetQueryOptionW https://docs.microsoft.com/en-us/windows/win32/api/wininet/nf-wininet-internetqueryoptionw • InternetCrackUrlA https://docs.microsoft.com/en-us/windows/win32/api/wininet/nf-wininet-internetcrackurla • WinHttpGetProxyForUrl https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpgetproxyforurl • WinHttpDetectAutoProxyConfigUrl https://docs.microsoft.com/en-us/windows/win32/api/winhttp/nf-winhttp-winhttpdetectautoproxyconfigurl • WlanQueryInterface https://docs.microsoft.com/en-us/windows/win32/api/wlanapi/nf-wlanapi-wlanqueryinterface

API Category	Windows APIs
	• WlanCloseHandle https://docs.microsoft.com/en-us/windows/win32/api/wlanapi/nf-wlanapi-wlanclosehandle • WlanOpenHandle https://docs.microsoft.com/en-us/windows/win32/api/wlanapi/nf-wlanapi-wlanopenhandle • WlanFreeMemory https://docs.microsoft.com/en-us/windows/win32/api/wlanapi/nf-wlanapi-wlanfreememory • WlanGetProfile https://docs.microsoft.com/en-us/windows/win32/api/wlanapi/nf-wlanapi-wlangetprofile • GetAdaptersAddresses https://docs.microsoft.com/en-us/windows/win32/api/iphlpapi/nf-iphlpapi-getadaptersaddresses
SOAP http	System.Web.Services.Protocols.SoapHttpClientProtocol methods • Invoke https://docs.microsoft.com/en-us/dotnet/api/system.web.services.protocols.soaphttpclientprotocol.invoke • BeginInvoke https://docs.microsoft.com/en-us/dotnet/api/system.web.services.protocols.soaphttpclientprotocol.begininvoke • EndInvoke https://docs.microsoft.com/en-us/dotnet/api/system.web.services.protocols.soaphttpclientprotocol.endinvoke • InvokeAsync https://docs.microsoft.com/en-us/dotnet/api/system.web.services.protocols.soaphttpclientprotocol.invokeasync
Windows Service	• OpenSCManagerW https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-openscmanagerw • OpenServiceW https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-openservicew • QueryServiceStatus https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-queryservicestatus • CloseServiceHandle https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-closeservicehandle • StartServiceW https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-startservicew • ControlService https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-controlservice • QueryServiceConfigW https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-queryserviceconfigw • DeleteService https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-deleteservice

API Category	Windows APIs
	• EnumServicesStatusW https://docs.microsoft.com/en-us/windows/win32/api/winsvc/nf-winsvc-enumservicesstatusw
Microsoft Certificate Store	• CertAddCertificateContextToStore https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-certaddcertificatecontexttostore • CertCloseStore https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-certclosestore • CertComparePublicKeyInfo https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-certcomparepublickeyinfo • CertEnumCertificatesInStore https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-certenumcertificatesinstore • CertEnumCRLsInStore https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-certenumcrlsinstore • CertFindCertificateInStore https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-certfindcertificateinstore • CertNameToStr https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-certnametostra • CertOpenStore https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-certopenstore • CertOpenSystemStore https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-certopensystemstorea • CertSetCertificateContextProperty https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-certsetcertificatecontextproperty • PFXExportCertStore https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-pfxexportcertstore
Microsoft CryptoAPI	• CryptAcquireContext https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptacquirecontexta • CryptBinaryToString https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptbinarytostringa • CryptCreateHash https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptcreatehash • CryptDecryptMessage https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptdecryptmessage • CryptDeriveKey https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptderivekey

API Category	Windows APIs
	• CryptEncrypt https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptencrypt • CryptDestroyHash https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptdestroyhash • CryptDestroyKey https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptdestroykey • CryptExportPublicKeyInfo https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptexportpublickeyinfo • CryptGetUserKey https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptgetuserkey • CryptHashData https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-crypthashdata • CryptImportKey https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptimportkey • CryptMsgOpenToDecode https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptmsgopentodecode • CryptMsgUpdate https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptmsgupdate • CryptMsgGetParam https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptmsggetparam • CryptReleaseContext https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptreleasecontext • CryptGenRandom https://docs.microsoft.com/en-us/windows/win32/api/wincrypt/nf-wincrypt-cryptgenrandom • BCryptGenRandom https://docs.microsoft.com/en-us/windows/win32/api/bcrypt/nf-bcrypt-bcryptgenrandom

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_API_EXT.1-ATE-01

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.

Summary

The evaluator found documentation for all APIs listed in [ST] during evaluation of the Security Target. Please refer to Table 4 in the ASE report.

2.3.6.3 Software Identification and Versions (FPT_IDV_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FPT_IDV_EXT.1-ASE-01

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

Summary

[ST] specifies the assignment of "other version information" for the selection in FPT_IDV_EXT.1 and specifies "Product Name" and "Product Version" file properties. Section 7.1.6.4, "TOE Identification," explains how the TOE is identified and how the Product Name and Product Version can be found.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_IDV_EXT.1-ATE-01

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.

Summary

Version information was found during the test for FPT_TUD_EXT.1.2. SWID tags are not used by the TOE and not selected in [ST].

2.3.6.4 Use of Third Party Libraries (FPT_LIB_EXT.1)

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_LIB_EXT.1-ATE-01

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.

Summary

The evaluator mapped the dynamic libraries found in the application installation directory against the list of 3rd-party libraries provided in the TSS and confirmed that all DLLs are part of a listed 3rd-party component.

Some 3rd-party libraries are not maintained by the 3rd-party because the TOE developer has incorporated a snapshot (i.e. fork) of the code into the source code for the TOE. The TOE developer maintains and modifies this code as needed and considers it internal TOE component code.

2.3.6.5 Integrity for Installation and Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.1.1-AGD-01

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

Summary

Section 6.2 of the [CC-CFG], *How to Check for Updates*, describes how to check updates for the TOE. Section 7 of the [CC-CFG], *Verify Authenticity of the Install Package*, describes how to verify the TOE downloadable package/updates.

Test Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.1.1-ATE-01

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.

Summary

The evaluator performed the update check as per the procedure defined in the operational guidance [CC-CFG] which found no update available and did not produce any errors.

FPT_TUD_EXT.1.2

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.1.2-AGD-01

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

Summary

The evaluator reviewed and determined that section 6.3, *Cloud Extender Versioning*, of the [CC-CFG] describes how to view the version of the TOE with either of the following two methods.

1. Go to **Control Panel»Programs and Features**.
2. Locate the **Cloud Extender** in the list.
3. The version will be displayed in the far right **Version** column.

or

1. Open the File Manager.
2. Navigate to the `C:\Program Files(x86)\MaaS360\Cloud Extender` folder.
3. Right-click on the `emsagent.exe` file.
4. Select **Properties**.
5. Select the **Details** tab to see the version.

Test Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.1.2-ATE-01

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.

Summary

The evaluator successfully checked the application version using the procedure documented in the operational guidance [CC-CFG] (AGD_OPE.1) and verified that the version matched.

FPT_TUD_EXT.1.3

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.1.3-ATE-01

[TD0548] For iOS: The evaluator shall consider the requirement met because the platform forces applications to write all data within the application working directory (sandbox).

[TD0548] For all other platforms: 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.

Summary

The evaluator installed the application but did not make any configuration changes using the Configuration Tool (ConfigTool.exe). The evaluator then created a hash of all binaries in the TOE installation directory. The evaluator then launched the Configuration Tool and exercised all features

of the TOE modules as documented in the TSS and operational guidance [CC-CFG]. The evaluator then validated the TOE binaries against the previously generated hash and verified that no changes were made.

FPT_TUD_EXT.1.4

TSS Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.1.4-ASE-01

[TD0561] The evaluator shall verify that the TSS identifies how the application 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.

Summary

[ST] section 7.1.6.5.1, "Security Update Process for the Cloud Extender," states the TOE installation package is signed by IBM in accordance with the Microsoft Authenticode process using a Symantec certificate issued to IBM. This section also explains that automatic updates are not available for the TOE and that to update to a new version, a new installer must be obtained from the customer portal.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

FPT_TUD_EXT.1.5

TSS Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.1.5-ASE-01

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"/>.

Summary

[ST] section 7.1.6.5.1, "Security Update Process for the Cloud Extender," explains the installer must be obtained from the customer portal. It goes on to explain that an update is merely a new install and therefore the evaluator concludes the description of how the installation works applies to updates.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.6.6 Integrity for Installation and Update (FPT_TUD_EXT.2)

FPT_TUD_EXT.2.1

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.2.1-ATE-01

The evaluator shall verify that application updates are distributed in the format supported by the platform. This varies per platform:

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

For 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/en-us/library/ms537364(v=vs.85).aspx for details regarding Authenticode signing.*

For iOS: *The evaluator shall ensure that the application is packaged in the IPA format.*

For 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.*

For Solaris: *The evaluator shall ensure that the application is packaged in the PKG format.*

For macOS: *The evaluator shall ensure that application is packaged in the DMG format, the PKG format, or the MPKG format.*

Summary

The evaluator examined the provided application installer which was packaged in .EXE format and verified Authenticode signing using the Microsoft SignTool utility.

FPT_TUD_EXT.2.2

TSS Assurance Activities

No assurance activities defined.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.2.2-ATE-01

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

For 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 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.*

Summary

The evaluator created a manifest of the entire filesystem prior to installation. After other testing activities were completed the application was uninstalled and a second manifest of the entire filesystem was created. These manifests were then compared and no files including configuration, output, or audit/log related to the TOE were found.

FPT_TUD_EXT.2.3

TSS Assurance Activities

Assurance Activity AA-FPT_TUD_EXT.2.3-ASE-01

[TD0561] 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.

Summary

[ST] sections 7.1.6.5.2, "Security Update Process for the Cloud Extender," describes how the Cloud Extender installers are signed by IBM in accordance with the Microsoft Authenticode process using a Class 3 SHA-256 certificate provided to IBM by Symantec. It is further stated that the signature is the only authorized source for the TOE.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

No assurance activities defined.

2.3.7 Trusted path/channels (FTP)

2.3.7.1 Protection of Data in Transit (FTP_DIT_EXT.1)

TSS Assurance Activities

Assurance Activity AA-FTP_DIT_EXT.1-ASE-01

[TD0601] For platform-provided functionality, the evaluator shall verify the TSS contains the calls to the platform that TOE is leveraging to invoke the functionality.

Summary

[ST] section 7.1.7, "Trusted path/channels," explains the TOE uses HTTPS and TLS for this purpose. This section lists the categories of platform API functions providing this protection and refers to the table in section 7.1.6.2 provided for FPT_API_EXT.1 for the specific APIs for each category.

Guidance Assurance Activities

No assurance activities defined.

Test Assurance Activities

Assurance Activity AA-FTP_DIT_EXT.1-ATE-01

[TD0601] 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.*

For Android: *If "not transmit any data" is selected, the evaluator shall ensure that the application's AndroidManifest.xml file does not contain a uses-permission or uses-permission-sdk-23 tag containing android:name="android.permission.INTERNET". In this case, it is not necessary to perform the above Tests 1, 2, or 3, as the platform will not allow the application to perform any network communication.*

For iOS: *If "encrypt all transmitted data" is selected, the evaluator shall ensure that the application's Info.plist file does not contain the NSAllowsArbitraryLoads or NSExceptionAllowsInsecureHTTPLoads keys, as these keys disable iOS's Application Transport Security feature.*

Summary

The evaluator established all trusted channels to the operational environment components documented in the [ST] as well as operational guidance ([CC-CFG]) using known credentials. Packet capture traffic was collected.

Test 1: The connections were confirmed to use encrypted protocols as described in the TSS.

Test 2: A string search was performed on the known credentials and none was found in plain text.

Test 3: No user credentials are transmitted by the TOE in plaintext.

2.4 Security Assurance Requirements

2.4.1 Guidance documents (AGD)

2.4.1.1 Operational user guidance (AGD_OPE.1)

Assurance Activity AA-AGD_OPE.1-AGD-01

Some of the contents of the operational guidance will be verified by the assurance activities in the Security Functional Requirements 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.

Summary

Section 1.3, *Cryptographic Acknowledgment*, of the [CC-CFG] specifies that cryptographic services are provided by both the Windows platform and OpenSSL and that only these two cryptographic services were tested as part of the evaluated configuration. The TOE is bound to OpenSSL, and the cryptographic functionality cannot be configured or modified.

Section 2.1, *Make TLS 1.2 the System Default on Windows Server 2019*, of the [CC-CFG] describes how to configure the Windows platform-provided cryptographic functinonality during installation of the TOE as folows:

There are three exported registry settings, described in Appendix A that can be used to create .reg files to run on the Cloud Extender server. After running these three files you can open the registry editor to view the changes required to limit the protocol to TLS 1.2 and specific ciphers.

Perform the following steps.

- Create three .reg files as described in Appendix A: Protocols.reg, CipherAvail.reg, and CCCiphers.reg.
- Use Remote Desktop to access the Cloud Extender server and copy these three files to a temp folder.
- Run each of the files and select Yes to the prompt Are you sure you want to modify the registry…
- Reboot the server after running all three reg files. The changes will not take effect until after a reboot

Please see AGD_ASPP.1-6 for the provided information regarding how to check for updates for the TOE.

Section 7, *Verify Authenticity of the Install Package*, of the [CC-CFG] describes how to verify the authenticity of the installation package, which is signed using a Symantec certificate issued to IBM.

1. Open File Manager and right click on the installer file.
2. Select Properties and choose the Digital Signatures tab.
3. Select one of the signature and click Details.
4. Click "View Certificate".
5. Verify that the code signing certificate was issued to IBM by a trusted root authority (such as DigiCert).

Section 1.6, *TOE Security Functionality*, of the [CC-CFG] describes the following security functionality provided by the TOE in the evaluated configuration as follows:

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

2.4.1.2 Preparative procedures (AGD_PRE.1)

Assurance Activity AA-AGD_PRE.1-AGD-01

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.

Summary

Section 1.5, *TOE Description*, of the [ST] describes the TOE as a software application that is installed and runs as a service on a Microsoft Windows operating system - specifically Microsoft Windows Server 2019 Standard version 1809 (x64) in this case.

The evaluator reviewed [CC-CFG] and determined that section 1.2, *Evaluated Versions*, specifies that the evaluation was performed using Microsoft Windows Server 2019 Standard version 1809 (x64).

2.4.2 Tests (ATE)

2.4.2.1 Independent testing - conformance (ATE_IND.1)

Assurance Activity AA-ATE_IND.1-ATE-01

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 evaluation activities.

While it is not necessary to have one test case per test listed in an evaluation 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 effect; 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 (e.g 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.

Summary

The evaluator prepared and executed the test plan to cover the Assurance Activities required by the protection profile. This test plan identifies the TOE platforms, test prerequisites, steps for each test, expected results, and actual observed results.

2.4.3 Life-cycle support (ALC)

2.4.3.1 Labelling of the TOE (ALC_CMC.1)

Assurance Activity AA-ALC_CMC.1-ALC-01

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.

Summary

The developer provided [ST] which is the Security Target for the evaluation. [ST] section 1.2 *TOE Identification* identifies the TOE as "IBM Maas360 v2.106.500.016 Cloud Extender" with the following modules, which are identified in section 1.5.

- Certificate Authority
- Exchange Integration for Managing Active Sync Devices
- Corporate User Visibility
- Corporate Directory Authentication

IBM MaaS360 Cloud Extender Donfiguration Tool version 2.106.500.016

In short, the TOE is uniquely identified as "IBM MaaS360 v2.106.500.016 Cloud Extender"

The developer provided [CC-CFG], *MaaS360 Cloud Extender Common Criteria Guide*, and [ADM_GUIDE], *MaaS360 Cloud Extender Admin Guide*, which are the guidance documents covering the secure preparation, configuration, and operation of the TOE in the evaluated configuration.

2.4.3.2 TOE CM coverage (ALC_CMS.1)

Assurance Activity AA-ALC_CMS.1-ALC-01

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.

Summary

As stated in the previous assurance activity, the TOE is uniquely identified in the Security Target [ST] and guidance documentation. Also, The TSF defined in [ST] is drawn directly from the Protection Profile [ASPPv1.3], thus is implicitly uniquely identified.

Additionally, [CC-CFG] section 1.2, *Evaluated Versions*, explicitly states the TOE's platform as Microsoft Windows Server 2019 version 1809 (x64).

Assurance Activity AA-ALC_CMS.1-ALC-02

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.

Summary

The TOE is not capable of having functional components added by an end user, so the requirement for guidance regarding development environments is not applicable.

As stated in the previous assurance activity, the TOE is uniquely identified in the Security Target [ST] and guidance documentation. Also, The TSF defined in [ST] is drawn directly from the Protection Profile [ASPPv1.3], thus is implicitly uniquely identified.

Additionally, [CC-CFG] section 1.2, *Evaluated Versions*, explicitly states the TOE's platform as Microsoft Windows Server 2019 version 1809 (x64).

2.4.4 Vulnerability assessment (AVA)

2.4.4.1 Vulnerability survey (AVA_VAN.1)

Assurance Activity AA-AVA_VAN.1-AVA-01

[TD0554] 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.

[TD0554] The evaluator documents the sources consulted and the vulnerabilities found in the report.

[TD0554] 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.

[TD0554] ***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.*

Summary

AVA_VAN.1 (pp_app_v1.3) *The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.*

The TOE provides guidance documents and an install package Windows executable. When installed, the executable and required support libraries (DLL files) are easily accessible in the Windows filesystem.

The TOE is installed and runs on Windows Server 2019. The evaluator installed the TOE according to the procedure described in [CC-CFG].

AVA_VAN.2 (pp_app_v1.3) *The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE*

The evaluator searched the Common Vulnerabilities and Exposures (CVE) database at https://cve.mitre.org/cve/cve.html for relevant vulnerabilities. The search terms used were determined based on the component list provided in [ST] and the TOE product name. The evaluator included the following items in the public search:

- IBM MaaS360 Cloud Extender
- Microsoft Windows Server 2019 Standard version 1809 (x64)
- TLS 1.2
- XMPP (Extensible Messaging and Presence Protocol)
- Exchange Integration for Active Sync Devices Module
- Corporate Directory Authentication Modules
- Corporate User Visibility Module
- Certificate Authority Module
- Windows Cryptography API: Next Generation
- .NET 4.7.2

The evaluator also searched the following public websites for known vulnerabilities:

- IBM Vendor Support Page https://www.ibm.com/mysupport
- OpenSSL vulnerability page https://www.openssl.org/news/vulnerabilities.html
- CISA vulnerability catalog https://www.cisa.gov/known-exploited-vulnerabilities-catalog

The search was performed in the following time frames:

- April 20-22, 2021
- August 22-24, 2021
- December 12-14, 2021
- June 20-22, 2022
- July 15-19, 2022
- July 25-29, 2022
- August 08-09, 2022
- September 12, 2022

No vulnerabilities applicable to the TOE in the evaluated configuration were found in this search.

AVA_VAN.3 (pp_app_v1.3) *The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.*

As shown in AVA_VAN.1-5 through AVA_VAN.1-8, the evaluator ran a port scan on the TOE using nmap version 7.92. Only expected ports were found to be open.

The evaluator also ran a virus scan on the TOE with Windows Defender. Windows Defender is packaged with Windows Server 2019 and has frequent updates to its definitions. The scan found no viruses, and thus the evaluator considers the TOE to be free of malicious files in this configuration.

A Appendixes

A.1 References

ADM_GUIDE	**MaaS360 Cloud Extender Admin Guide**	
	Version	1.0
	Date received	2022-07-20
	File name	agd/IBM_MaaS360_CE_Admin_Guide_v1.0.pdf
ASPPv1.3	**Protection Profile for Application Software Version 1.3**	
	Version	1.3
	Date	2016-02-22
	Location	https://www.niap-ccevs.org/MMO/pp/pp_app_v1.3.pdf
CC	**Common Criteria for Information Technology Security Evaluation**	
	Version	3.1R5
	Date	April 2017
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf
CC-CFG	**MaaS360 Cloud Extender Common Criteria Guide**	
	Version	1.0
	Date received	2020-07-20
	File name	agd/IBM_MaaS360_CE_Common_Criteria_Guide_v1.0.pdf
CCEVS-TD0416	**Correction to FCS_RBG_EXT.1 Test Activity**	
	Date	2019-04-24
	Location	https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0416
CCEVS-TD0427	**Reliable Time Source**	
	Date	2019-06-11
	Location	https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0427
CCEVS-TD0434	**Windows Desktop Applications Test**	
	Date	2019-07-22
	Location	https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0434
CCEVS-TD0435	**Alternative to SELinux for FPT_AEX_EXT.1.3**	
	Date	2019-07-26
	Location	https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0435

CCEVS-TD0437 **Supported Configuration Mechanism**
Date 2019-07-23
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0437

CCEVS-TD0442 **Updated TLS Ciphersuites for TLS Package**
Date 2019-08-21
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0442

CCEVS-TD0445 **User Modifiable File Definition**
Date 2019-10-09
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0445

CCEVS-TD0465 **Configuration Storage for .NET Apps**
Date 2019-11-08
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0465

CCEVS-TD0469 **Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1**
Date 2019-11-20
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0469

CCEVS-TD0495 **FIA_X509_EXT.1.2 Test Clarification**
Date 2020-01-29
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0495

CCEVS-TD0498 **Application Software PP Security Objectives and Requirements Rationale**
Date 2020-01-31
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0498

CCEVS-TD0499 **Testing with pinned certificates**
Date 2020-02-04
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0499

CCEVS-TD0510 **Obtaining random bytes for iOS/macOS**
Date 2020-03-03
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0510

CCEVS-TD0513 **CA Certificate loading**
Date 2020-05-26
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0513

CCEVS-TD0515 **Use Android APK manifest in test**
Date 2020-06-08
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0515

CCEVS-TD0519 **Linux symbolic links and FMT_CFG_EXT.1**
Date 2020-06-18
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0519

CCEVS-TD0543 **FMT_MEC_EXT.1 evaluation activity update**
Date 2020-09-15
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0543

CCEVS-TD0544 **Alternative testing methods for FPT_AEX_EXT.1.1**
Date 2020-09-15
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0544

CCEVS-TD0548 **Integrity for installation tests in AppSW PP 1.3**
Date 2020-09-30
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0548

CCEVS-TD0554 **iOS/iPadOS/Android AppSW Virus Scan**
Date 2020-10-30
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0554

CCEVS-TD0561 **Signature verification update**
Date 2021-01-15
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0561

CCEVS-TD0582 **PP-Configuration for Application Software and Virtual Private Network (VPN) Clients now allowed**
Date 2021-04-16
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0582

CCEVS-TD0588 **Session Resumption Support in TLS package**
Date 2021-05-12
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0588

CCEVS-TD0598 **Expanded AES Modes in FCS_COP for App PP**
Date 2021-08-03
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0598

CCEVS-TD0600 **Conformance claim sections updated to allow for MOD_VPNC_V2.3**
Date 2021-08-10
Location https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0600

CCEVS-TD0601	**X.509 SFR Applicability in App PP**	
	Date	2021-09-22
	Location	https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0601
CEM	**Common Methodology for Information Technology Security Evaluation**	
	Version	3.1R5
	Date	April 2017
	Location	http://www.commoncriteriaportal.org/files/ccfiles/CEMV3.1R5.pdf
EAR	**MaaS360 Cloud Extender Entropy Assessment Report**	
	Version	2.2
	Date	June 24, 2022
	File name	agd/MaaS360CloudExtenderEAR.pdf
ST	**IBM MaaS360 2.106.500.016 Cloud Extender Security Target**	
	Version	1.4
	Date	2022-07-20
	File name	ase/IBM_Mass360_ST_1.4.pdf
TLSPKGv1.1	**Functional Package for Transport Layer Security (TLS)**	
	Version	1.1
	Date	2019-02-12
	Location	https://www.niap-ccevs.org/MMO/PP/PKG_TLS_V1.1.pdf

A.2 Glossary

Augmentation
The addition of one or more requirement(s) to a package.

Authentication data
Information used to verify the claimed identity of a user.

Authorised user
A user who may, in accordance with the SFRs, perform an operation.

Class
A grouping of CC families that share a common focus.

Component
The smallest selectable set of elements on which requirements may be based.

Connectivity
The property of the TOE which allows interaction with IT entities external to the TOE. This includes exchange of data by wire or by wireless means, over any distance in any environment or configuration.

Dependency
A relationship between components such that if a requirement based on the depending component is included in a PP, ST or package, a requirement based on the component that is depended upon must normally also be included in the PP, ST or package.

Deterministic RNG (DRNG)
An RNG that produces random numbers by applying a deterministic algorithm to a randomly selected seed and, possibly, on additional external inputs.

Element
An indivisible statement of security need.

Entropy
The entropy of a random variable X is a mathematical measure of the amount of information gained by an observation of X.

Evaluation
Assessment of a PP, an ST or a TOE, against defined criteria.

Evaluation Assurance Level (EAL)
An assurance package, consisting of assurance requirements drawn from CC Part 3, representing a point on the CC predefined assurance scale.

Evaluation authority
A body that implements the CC for a specific community by means of an evaluation scheme and thereby sets the standards and monitors the quality of evaluations conducted by bodies within that community.

Evaluation scheme
The administrative and regulatory framework under which the CC is applied by an evaluation authority within a specific community.

Exact conformance
a subset of Strict Conformance as defined by the CC, is defined as the ST containing all of the requirements in the Security Requirements section of the PP, and potentially requirements from Appendices of the PP. While iteration is allowed, no additional requirements (from the CC parts 2 or 3) are allowed to be included in the ST. Further, no requirements in the Security Requirements section of the PP are allowed to be omitted.

Extension
The addition to an ST or PP of functional requirements not contained in Part 2 and/or assurance requirements not contained in Part 3 of the CC.

External entity
Any entity (human or IT) outside the TOE that interacts (or may interact) with the TOE.

Family
A grouping of components that share a similar goal but may differ in emphasis or rigour.

Formal
Expressed in a restricted syntax language with defined semantics based on well-established mathematical concepts.

Guidance documentation
Documentation that describes the delivery, preparation, operation, management and/or use of the TOE.

Identity
A representation (e.g. a string) uniquely identifying an authorised user, which can either be the full or abbreviated name of that user or a pseudonym.

Informal
Expressed in natural language.

Object
A passive entity in the TOE, that contains or receives information, and upon which subjects perform operations.

Operation (on a component of the CC)
Modifying or repeating that component. Allowed operations on components are assignment, iteration, refinement and selection.

Operation (on an object)
A specific type of action performed by a subject on an object.

Operational environment
The environment in which the TOE is operated.

Organisational Security Policy (OSP)
A set of security rules, procedures, or guidelines imposed (or presumed to be imposed) now and/or in the future by an actual or hypothetical organisation in the operational environment.

Package
A named set of either functional or assurance requirements (e.g. EAL 3).

PP evaluation
Assessment of a PP against defined criteria.

Protection Profile (PP)
An implementation-independent statement of security needs for a TOE type.

Random number generator (RNG)
A group of components or an algorithm that outputs sequences of discrete values (usually represented as bit strings).

Refinement
The addition of details to a component.

Role
A predefined set of rules establishing the allowed interactions between a user and the TOE.

Secret
Information that must be known only to authorised users and/or the TSF in order to enforce a specific SFP.

Secure state
A state in which the TSF data are consistent and the TSF continues correct enforcement of the SFRs.

Security attribute
A property of subjects, users (including external IT products), objects, information, sessions and/or resources that is used in defining the SFRs and whose values are used in enforcing the SFRs.

Security Function Policy (SFP)
A set of rules describing specific security behaviour enforced by the TSF and expressible as a set of SFRs.

Security objective
A statement of intent to counter identified threats and/or satisfy identified organisation security policies and/or assumptions.

Security Target (ST)
An implementation-dependent statement of security needs for a specific identified TOE.

Seed
Value used to initialize the internal state of an RNG.

Selection
The specification of one or more items from a list in a component.

Semiformal
Expressed in a restricted syntax language with defined semantics.

ST evaluation
Assessment of an ST against defined criteria.

Subject
An active entity in the TOE that performs operations on objects.

Target of Evaluation (TOE)
A set of software, firmware and/or hardware possibly accompanied by guidance.

TOE evaluation
Assessment of a TOE against defined criteria.

TOE resource
Anything useable or consumable in the TOE.

TOE Security Functionality (TSF)
A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the SFRs.

Transfers outside of the TOE
TSF mediated communication of data to entities not under control of the TSF.

True RNG (TRNG)
A device or mechanism for which the output values depend on some unpredictable source (noise source, entropy source) that produces entropy.

Trusted channel
A means by which a TSF and a remote trusted IT product can communicate with necessary confidence.

Trusted path
A means by which a user and a TSF can communicate with necessary confidence.

TSF data
Data created by and for the TOE, that might affect the operation of the TOE.

TSF Interface (TSFI)
A means by which external entities (or subjects in the TOE but outside of the TSF) supply data to the TSF, receive data from the TSF and invoke services from the TSF.

User
See external entity

User data
Data created by and for the user, that does not affect the operation of the TSF.