Assurance Activity Report for
Trellix Endpoint Security (HX) Agent v35.31.31

Trellix Endpoint Security (HX) Agent v35.31.31 Security Target
Version 2.3

Protection Profile for Application Software, Version 1.4

Functional Package for Transport Layer Security (TLS), Version 1.1

AAR Version 1.4, May 29, 2024

Evaluated by:

2400 Research Blvd, Suite 395
Rockville, MD 20850

Prepared for:

National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:
Trellix US LLC,
601 McCarthy Blvd.,
Milpitas, CA 95035

The Author of the Security Target:
Acumen Security, LLC.

The TOE Evaluation was Sponsored by:
Trellix US LLC,

Evaluation Personnel:
Ruban Abinesh
Akshay Jain
Jonathan Anglin
Fathi Nasraoui
Acumen Security, LLC.

Common Criteria Version
Common Criteria Version 3.1 Revision 5

Common Evaluation Methodology Version
CEM Version 3.1 Revision 5

Revision History

VERSION	DATE	CHANGES
0.1	06/06/2023	Initial Release
0.2	06/29/2023	TSS Activities
0.3	07/10/2023	TSS updates
0.4	08/15/2023	Updates to TSS AAs based on internal review comments
0.5	09/19/2023	Updates to TSS, ATE, AGD and AVA AAs based on internal review comments
0.6	11/02/2023	QA AGD Review
0.7	12/01/2023	Changes to AGD verdicts after Review
0.8	02/28/2024	QA and updates to verdicts
0.9	03/19/2024	Updated section 6 and 7
1.0	04/18/2024	Addressed peer lead comments
1.1	05/10/2024	Addressing validators ECR comments
1.2	05/22/2024	New ST and AGD versions
1.3	05/23/2024	Addressing validators ECR comments
1.4	05/29/2024	AVA VAN search date update

Contents

1 TOE Overview 9
1.1 Cryptographic Support 9
2 Assurance Activities Identification 10
2.1 TOE Documentation 10
3 Technical Decisions 11
4 Test Bed Descriptions 12
4.1 Detailed Test Configuration 13
4.2 Test Time & Location 13
5 Detailed TSS and Guidance Evaluation Activities 14
5.1 TSS and Guidance Activities (Cryptographic Support) 14
5.1.1 FCS_CKM_EXT.1 14
5.1.1.1 FCS_CKM_EXT.1 TSS 1 [**TD0717**] 14
5.1.2 FCS_CKM.1/AK 14
5.1.2.1 FCS_CKM.1/AK TSS 1 **[TD0717]** 14
5.1.2.2 FCS_CKM.1/AK Guidance 1 **[TD0717]** 14
5.1.3 FCS_CKM.2 15
5.1.3.1 FCS_CKM.2 TSS 1 **[TD0717]** 15
5.1.3.2 FCS_CKM.2 TSS 2 [**TD0717**] 15
5.1.3.3 FCS_CKM.2 Guidance 1 **[TD0717]** 15
5.1.4 FCS_COP.1/SKC 16
5.1.4.1 FCS_COP.1/SKC Guidance 1 **[TD0717]** 16
5.1.5 FCS_COP.1/Hash 16
5.1.5.1 FCS_COP.1/Hash TSS 1 **[TD0717]** 16
5.1.6 FCS_HTTPS_EXT.1/Client 16
5.1.6.1 FCS_HTTPS_EXT.1.1/Client TSS 1 16
5.1.7 FCS_RBG_EXT.1 17
5.1.7.1 FCS_RBG_EXT.1 TSS 1 17
5.1.7.2 FCS_RBG_EXT.1 TSS 2 17
5.1.7.3 FCS_RBG_EXT.1 TSS 3 17
5.1.8 FCS_RBG_EXT.2 17
5.1.8.1 FCS_RBG_EXT.2.2 TSS 1 17
5.1.9 FCS_STO_EXT.1 18
5.1.9.1 FCS_STO_EXT.1 TSS 1 18
5.1.10 FCS_TLS_EXT.1 18
5.1.10.1 FCS_TLS_EXT.1 Guidance 1 18
5.1.11 FCS_TLSC_EXT.1 18
5.1.11.1 FCS_TLSC_EXT.1.1 TSS 1 **[TD0442]** 18
5.1.11.2 FCS_TLSC_EXT.1.1 Guidance 1 **[TD0442]** 19
5.1.11.3 FCS_TLSC_EXT.1.2 TSS 1 **[TD0499]** 19
5.1.11.4 FCS_TLSC_EXT.1.2 TSS 2 **[TD0499]** 20
5.1.11.5 FCS_TLSC_EXT.1.2 Guidance 1 [**TD0499**] 20
5.1.11.6 FCS_TLSC_EXT.1.3 TSS 1 20
5.1.12 FCS_TLSC_EXT.2 21
5.1.12.1 FCS_TLSC_EXT.2 TSS 1 21
5.1.12.2 FCS_TLSC_EXT.2 Guidance 1 21

5.2 TSS and Guidance Activities (User Data Protection) 22
5.2.1 FDP_DAR_EXT.1 22
5.2.1.1 FDP_DAR_EXT.1 TSS 1 22
5.2.1.2 FDP_DAR_EXT.1 TSS 2 22
5.2.2 FDP_DEC_EXT.1 22
5.2.2.1 FDP_DEC_EXT.1.1 Guidance 1 22
5.2.2.2 FDP_DEC_EXT.1.1 Guidance 2 23
5.2.2.3 FDP_DEC_EXT.1.2 Guidance 1 23
5.2.2.4 FDP_DEC_EXT.1.2 Guidance 2 23
5.3 TSS and Guidance Activities (Identification and Authentication) 24
5.3.1 FIA_X509_EXT.1 24
5.3.1.1 FIA_X509_EXT.1.1 TSS 1 24
5.3.2 FIA_X509_EXT.2 25
5.3.2.1 FIA_X509_EXT.2.1 TSS 1 25
5.3.2.2 FIA_X509_EXT.2.1 TSS 2 25
5.4 TSS and Guidance Activities (Security Management) 25
5.4.1 FMT_CFG_EXT.1 25
5.4.1.1 FMT_CFG_EXT.1.1 TSS 1 25
5.4.2 FMT_MEC_EXT.1 26
5.4.2.1 FMT_MEC_EXT.1 TSS 1 26
5.4.2.2 FMT_MEC_EXT.1 TSS 2 26
5.4.3 FMT_SMF.1 27
5.4.3.1 FMT_SMF.1 Guidance 1 27
5.5 TSS and Guidance Activities (Privacy) 28
5.5.1 FPR_ANO_EXT.1 28
5.5.1.1 FPR_ANO_EXT.1 TSS 1 28
5.6 TSS and Guidance Activities (Protection of the TSF) 28
5.6.1 FPT_AEX_EXT.1 28
5.6.1.1 FPT_AEX_EXT.1.1 TSS 1 **[TD0798]** 28
5.6.1.2 FPT_AEX_EXT.1.5 TSS 1 [**TD0815**] 28
5.6.2 FPT_API_EXT.1 29
5.6.2.1 FPT_API_EXT.1 TSS 1 29
5.6.3 FPT_IDV_EXT.1 29
5.6.3.1 FPT_IDV_EXT.1 TSS 1 29
5.6.4 FPT_TUD_EXT.1 29
5.6.4.1 FPT_TUD_EXT.1.1 Guidance 1 29
5.6.4.2 FPT_TUD_EXT.1.2 Guidance 1 30
5.6.4.3 FPT_TUD_EXT.1.4 TSS 1 30
5.6.4.4 FPT_TUD_EXT.1.5 TSS 1 31
5.6.5 FPT_TUD_EXT.2 31
5.6.5.1 FPT_TUD_EXT.2.3 TSS 1 31
5.7 TSS and Guidance Activities (Trusted Path/Channels) 31
5.7.1 FTP_DIT_EXT.1 31
5.7.1.1 FTP_DIT_EXT.1 TSS 1 31
6 Detailed Testing Evaluation Activities 33
6.1 APP_V1.4 33
6.1.1 FCS_CKM.1/AK Test/CAVP 1 33
6.1.2 FCS_CKM.2 Test/CAVP 1 35

6.1.3 FCS_COP.1/SKC Test/CAVP 1 38
6.1.4 FCS_COP.1/Hash Test/CAVP 1 43
6.1.5 FCS_COP.1/Sig Test/CAVP 1 44
6.1.6 FCS_COP.1/KeyedHash Test/CAVP 1 45
6.1.7 FCS_RBG_EXT.2.1 Test/CAVP 1 45
6.1.8 FCS_HTTPS_EXT.1.1/Client Test #1 47
6.1.9 FCS_HTTPS_EXT.1.2/Client Test #1 47
6.1.10 FCS_HTTPS_EXT.1.3/Client Test #1 47
6.1.11 FCS_HTTPS_EXT.2.1 Test #1 48
6.1.12 FCS_RBG_EXT.1.1 48
6.1.13 FCS_RBG_EXT.2.2 49
6.1.14 FCS_STO_EXT.1.1 Test #1 49
6.1.15 FCS_STO_EXT.1.1 Test #2 49
6.1.16 FDP_DAR_EXT.1.1 Test #1 50
6.1.17 FDP_DAR_EXT.1.1 Test #2 50
6.1.18 FDP_DEC_EXT.1.1 Test #1 50
6.1.19 FDP_DEC_EXT.1.2 Test #1 51
6.1.20 FDP_NET_EXT.1.1 Test #1 51
6.1.21 FDP_NET_EXT.1.1 Test #2 52
6.1.22 FDP_NET_EXT.1.1 Test #3 52
6.1.23 FIA_X509_EXT.1.1 Test #1 52
6.1.24 FIA_X509_EXT.1.1 Test #2 54
6.1.25 FIA_X509_EXT.1.1 Test #3 55
6.1.26 FIA_X509_EXT.1.1 Test #4 56
6.1.27 FIA_X509_EXT.1.1 Test #5 57
6.1.28 FIA_X509_EXT.1.1 Test #6 58
6.1.29 FIA_X509_EXT.1.1 Test #7 59
6.1.30 FIA_X509_EXT.1.1 Test #8 59
6.1.31 FIA_X509_EXT.1.1 Test #9 60
6.1.32 FIA_X509_EXT.1.2 Test #1 60
6.1.33 FIA_X509_EXT.1.2 Test #2 61
6.1.34 FIA_X509_EXT.2.2 Test #1 61
6.1.35 FIA_X509_EXT.2.2 Test #2 62
6.1.36 FMT_CFG_EXT.1.1 Test #1 63
6.1.37 FMT_CFG_EXT.1.1 Test #2 63
6.1.38 FMT_CFG_EXT.1.1 Test #3 63
6.1.39 FMT_CFG_EXT.1.2 Test #1 63
6.1.40 FMT_MEC_EXT.1.1 Test #1 64
6.1.41 FMT_MEC_EXT.1.1 Test #2 64
6.1.42 FMT_SMF.1.1 Test #1 65
6.1.43 FPR_ANO_EXT.1.1 Test #1 65
6.1.44 FPT_AEX_EXT.1.1 Test #1 65
6.1.45 FPT_AEX_EXT.1.2 Test #1 66
6.1.46 FPT_AEX_EXT.1.3 Test #1 66
6.1.47 FPT_AEX_EXT.1.4 Test #1 67
6.1.48 FPT_AEX_EXT.1.5 Test #1 67

6.1.49 FPT_API_EXT.1.1 Test #1 68
6.1.50 FPT_IDV_EXT.1.1 Test #1 68
6.1.51 FPT_LIB_EXT.1.1 Test #1 68
6.1.52 FPT_TUD_EXT.1.1 Test #1 69
6.1.53 FPT_TUD_EXT.1.2 Test #1 69
6.1.54 FPT_TUD_EXT.1.3 Test #1 69
6.1.55 FPT_TUD_EXT.1.5 TSS #1 70
6.1.56 FPT_TUD_EXT.2.1 Test #1 70
6.1.57 FPT_TUD_EXT.2.2 Test #1 70
6.1.58 FTP_DIT_EXT.1.1 Test #1 71
6.1.59 FTP_DIT_EXT.1.1 Test #2 71
6.1.60 FTP_DIT_EXT.1.1 Test #3 71
6.2 TLSC-MA 72
6.2.1 FCS_TLSC_EXT.1 Test #1 72
6.2.2 FCS_TLSC_EXT.1 Test #2 72
6.2.3 FCS_TLSC_EXT.1 Test #3 73
6.2.4 FCS_TLSC_EXT.1 Test #4 73
6.2.5 FCS_TLSC_EXT.1 Test #5.1 73
6.2.6 FCS_TLSC_EXT.1 Test #5.2 74
6.2.7 FCS_TLSC_EXT.1 Test #5.3 74
6.2.8 FCS_TLSC_EXT.1 Test #5.4 74
6.2.9 FCS_TLSC_EXT.1 Test #5.5 74
6.2.10 FCS_TLSC_EXT.1 Test #5.6 75
6.2.11 FCS_TLSC_EXT.1 Test #5.7 75
6.2.12 FCS_TLSC_EXT.1 Test #6 75
6.2.13 FCS_TLSC_EXT.1 Test #7 76
6.2.14 FCS_TLSC_EXT.1 Test #8 77
6.2.15 FCS_TLSC_EXT.1 Test #9 77
6.2.16 FCS_TLSC_EXT.1 Test #10.1 78
6.2.17 FCS_TLSC_EXT.1 Test #10.2(a) 79
6.2.18 FCS_TLSC_EXT.1 Test #10.2(b) 80
6.2.19 FCS_TLSC_EXT.1 Test #10.2(c) 81
6.2.20 FCS_TLSC_EXT.1 Test #10.3(a) 82
6.2.21 FCS_TLSC_EXT.1 Test #10.3(b) 83
6.2.22 FCS_TLSC_EXT.1 Test #10.4 84
6.2.23 FCS_TLSC_EXT.1 Test #11 84
6.2.24 FCS_TLSC_EXT.1 Test #12 84
6.2.25 FCS_TLSC_EXT.1 Test #13a 85
6.2.26 FCS_TLSC_EXT.1 Test #13b 85
6.2.27 FCS_TLSC_EXT.1 Test #13c 85
6.2.28 FCS_TLSC_EXT.1 Test #14 85
6.2.29 FCS_TLSC_EXT.1 Test #15 86
6.2.30 FCS_TLSC_EXT.1 Test #16 86
6.2.31 FCS_TLSC_EXT.2 Test #1 86
6.2.32 FCS_TLSC_EXT.2 Test #2 87
6.2.33 FCS_TLSC_EXT.3.1 Test #1 87

6.2.34 FCS_TLSC_EXT.3.1 Test #2 87
6.2.35 FCS_TLSC_EXT.4.1 Test #1 87
6.2.36 FCS_TLSC_EXT.4.1 Test #2 88
6.2.37 FCS_TLSC_EXT.4.1 Test #3 88
6.2.38 FCS_TLSC_EXT.5.1 Test #1 89
7 Security Assurance Requirements 90
7.1 ADV: Developpment 90
7.1.1 ADV_FSP.1 Basic Functional Specification 90
7.1.1.1 ADV_FSP.1 TSS 1 90
7.2 AGD: Guidance Documentation 90
7.2.1 AGD_OPE.1 Operational User Guidance 90
7.2.1.1 AGD_OPE.1 Guidance 1 90
7.2.1.2 AGD_OPE.1 Guidance 2 90
7.2.2 AGD_PRE.1 Preparative Procedures 91
7.2.2.1 AGD_PRE.1 Guidance 1 91
7.3 ALC: Life-Cycle Support 92
7.3.1 ALC_CMC.1 Labelling of the TOE 92
7.3.1.1 ALC_CMC.1 TSS 1 92
7.3.1.2 ALC_CMC.1 TSS 2 92
7.3.1.3 ALC_CMC.1 Guidance 1 93
7.3.2 ALC_CMS.1 TOE CM Coverage 93
7.3.2.1 ALC_CMS.1 Guidance 1 93
7.3.2.2 ALC_CMS.1 Guidance 2 94
7.3.3 ALC_TSU_EXT.1 Timely Security Updates 94
7.3.3.1 ALC_TSU_EXT.1 TSS 1 94
7.3.3.2 ALC_TSU.1 TSS 2 95
7.3.3.3 ALC_TSU.1 TSS 3 95
7.4 ATE: Tests 95
7.4.1 ATE_IND.1 Independent Testing – Conformance 95
7.5 AVA: Vulnerability Assessment 96
7.5.1 AVA_VAN.1 Vulnerability Survey 96
7.5.1.1 AVA_VAN.1 Activity 1 **[Labgram #116]** 96
7.5.1.2 AVA_VAN.1 Activity 2 97
8 Conclusion 99

1 TOE Overview

The TOE is the Trellix Endpoint Security (HX) Agent v35.31.31, a software application residing on a host platform and interacting exclusively with a Trellix Endpoint Security (HX) Series appliance. The TOE is an enterprise-managed agent that runs in the background of the host platform of an endpoint to provide protection against common malware as well as advanced attack. Based on a defense in depth model, the TOE uses a modular architecture with default engines and downloadable modules to protect, detect and respond to security events. There are no users interacting with the TOE or being informed of any communication between the TOE and the HX Series appliance.

1.1 Cryptographic Support

The TOE implements cryptographic support for the following:

- TLS connectivity between itself and a Trellix Endpoint Security (HX) Series Appliance, including generation of 2048-bit RSA keys for a certificate signing request and implementation of all required cryptographic algorithms, and
- Digital certificate validation.
  The cryptographic algorithms the TOE implements and the CAVP certificate numbers are given in

Table 1. Each algorithm is implemented using the OpenSSL Cryptographic Library version 3.0.8 which is part of the TOE.

Table 1 TOE Cryptographic Algorithms and CAVP Certificate References

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
FCS_CKM.1/AK	RSA schemes using cryptographic key sizes of 2048-bit that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3	Trellix OpenSSL FIPS Provider v3.0.8	RSA KeyGen (FIPS186-4)	A5228
FCS_CKM.2	RSA key establishment schemes that meet the following: NIST Special Publication 800-56B, “Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography”	Trellix OpenSSL FIPS Provider v3.0.8	Vendor Affirmed	Vendor Affirmed
FCS_COP.1/ SKC	AES-CBC mode as defined in NIST SP 800-38A and cryptographic key sizes 128 bits and 256 bits	Trellix OpenSSL FIPS Provider v3.0.8	AES-CBC	A5228
FCS_COP.1/ Hash	SHA-1 and SHA-256 and message digest sizes 160 and 256 bits	Trellix OpenSSL FIPS Provider v3.0.8	SHA-1 SHA2-256	A5228
FCS_COP.1/ Sig	RSA scheme using cryptographic key sizes of 2048-bit that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5	Trellix OpenSSL FIPS Provider v3.0.8	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	A5228
FCS_COP.1/ KeyedHash	HMAC-SHA-1 and HMAC-SHA-256 with key sizes 256 and 160 bits used in HMAC and message digest sizes	Trellix OpenSSL FIPS Provider v3.0.8	HMAC-SHA-1 HMAC-SHA2- 256	A5228

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
	256 and 160 bits that meet the following: FIPS Pub 198-1,'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'			
FCS_RBG_EXT.2.1	An NIST Special Publication 800-90A using CTR_DRBG(AES) with a minimum of 256-bits	Trellix OpenSSL FIPS Provider v3.0.8	Counter DRBG	A5228

2 Assurance Activities Identification

The Assurance Activities contained within this document include all those defined within the Protection Profile for Application Software Version 1.4 and Functional Package for TLS Version 1.1 based upon the core SFRs and those implemented based on selections within the PPs/PKGs.

2.1 TOE Documentation

The following documents are essential to understanding and controlling the TOE in the evaluated configuration:

1. Trellix Endpoint Security (HX) Agent v35.31.31 Security Target, version 2.3 May 21, 2024 [ST]
2. Trellix Endpoint Security (HX) Agent v35.31.31 Common Criteria Guidance Supplement, version 1.4, May 21, 2024 [AGD]
3. Endpoint Security xAgent Deployment Guide Release 35.31.0, year 2023 [DG]

3 Technical Decisions

The following technical decisions were applied for this evaluation:

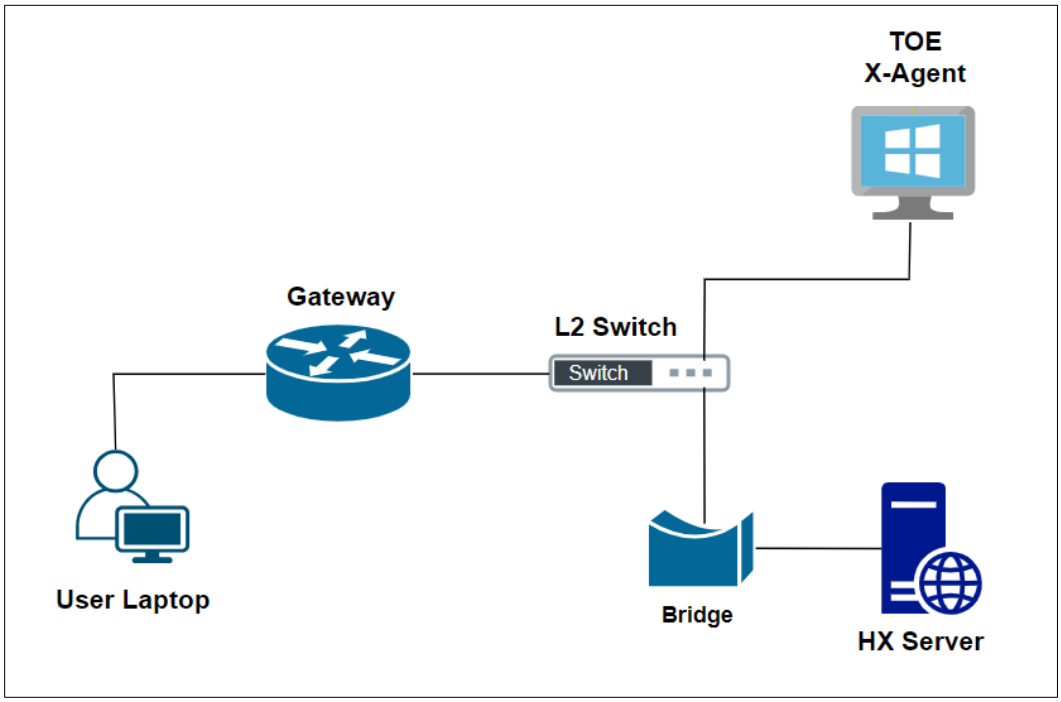
Table 2: Relevant technical decisions applicable to the TOE/ST.

Technical Decision	Applicable	Exclusion Rationale (where applicable)
PP_APP_v1.4: Active Related Technical Decisions		
0823 – Update to Microsoft Windows Exploit Protection link in FPT_AEX_EXT.1.3	Yes	
0822 – Correction to Windows Manifest File for FDP_DEC_EXT.1	Yes	
TD0815: Addition of Conditional TSS Activity for FPT_AEX_EXT.1.5	Yes	
TD0798: Static Memory Mapping Exceptions	Yes	
TD0780: FIA_X509_EXT.1 Test 4 Clarification	Yes	
TD0756 – Update for platform-provided full disk encryption	Yes	
TD0747: Configuration Storage Option for Android	No	TOE is based on Windows platform
TD0743: FTP_DIT_EXT.1.1 Selection exclusivity	Yes	
TD0736: Number of elements for iterations of FCS_HTTPS_EXT.1	No	Toe does not claim FCS_HTTPS_EXT.1/Server
TD0719: ECD for PP APP V1.3 and 1.4	Yes	
TD0717: Format changes for PP_APP_V1.4	Yes	
TD0664: Testing activity for FPT_TUD_EXT.2.2	Yes	
TD0650: Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4	No	ST does not claim PP-Module for VPN Clients, Version 2.4
TD0628: Addition of Container Image to Package Format	Yes	
PKG_TLS_v1.1: Active Related Technical Decisions		
TD0779: Updated Session Resumption Support in TLS package V1.1	No	ST does not claim TLS server
TD0770: TLSS.2 connection with no client cert	No	ST does not claim TLS server
TD0739: PKG_TLS_V1.1 has 2 different publication dates	No	ST does not claim TLS server
TD0726: Corrections to (D)TLSS SFRs in TLS 1.1 FP	No	ST does not claim TLS server
TD0513: CA Certificate loading	Yes	
TD0499: Testing with pinned certificates	Yes	
TD0469: Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	No	The TOE does not implement TLS Server
TD0442: Updated TLS Ciphersuites for TLS Package	Yes	

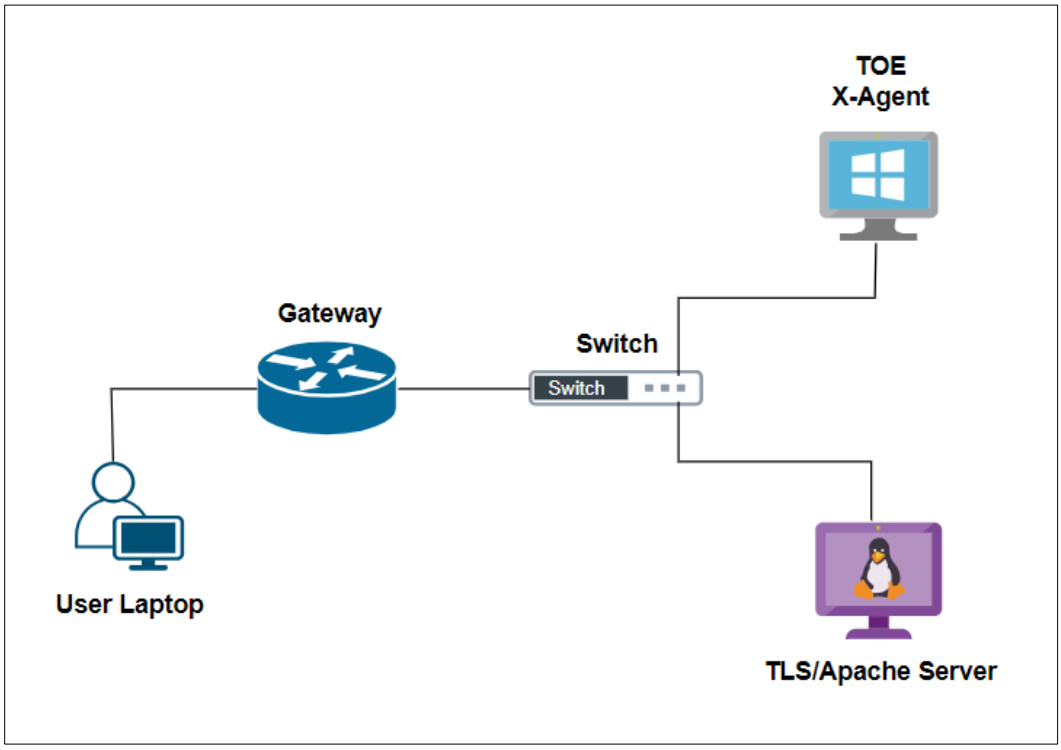
4 Test Bed Descriptions

The following is a visual representation of the test beds which were used for testing :

APP_v1.4



PKG_TLSC / X509



4.1 Detailed Test Configuration

Name	OS	Version	Function	Protocols	Time	Tools
X-Agent	Windows Server 2012 R2 Windows Server 2019 Windows 10 32-bits Windows 10 64-bits Windows 11	v35.31.31	TOE	RDP	Manually set and verified	Python v3.11.4, Microsoft Binscope 2014, HashMyFiles v2.43, WinMerge v2.16.30.0, EMET v5.5.5871.31892
HX Server	Red Hat Enterprise Linux 7 (64-bit)	v5.3.0.976321	External HX Series appliance/ TLS Server	HTTPS, SSH, TLS	Manually set and verified	N/A
Bridge	Ubuntu Linux (64-bit)	Ubuntu 20.04.6 LTS	Network Bridge	SSH	Manually set and verified	N/A
TLS Server	Ubuntu Linux (64-bit)	Ubuntu 20.04.6 LTS	TLS Server	TLS, SSH	Manually set and verified	OpenSSL v1.1.1f, acumen-tlsc, tcpdump v4.9.3
Apache Server	Ubuntu Linux (64-bit)	Apache/2.4.41 (Ubuntu)	Apache server/ CRL server	TLS, SSH	Manually set and verified	N/A
Switch	N/A	N/A	L2 Switch	N/A	N/A	N/A
Gateway	N/A	N/A	Gateway	N/A	N/A	N/A
HP Laptop	Windows 10	v22H2	Management Workstation	RDP	Manually set and verified	XCA v2.1.1, Putty v0.76, MobaXterm v21.3, Remote Desktop Connection v10.0.19041

Table 1: Testing configuration Table

4.2 Test Time & Location

All testing was carried out on the TOE running on Microsoft operating systems Windows Server 2012 R2, Windows Server 2019, Windows 10 32-bits, Windows 10 64-bits and Windows 11 hosted on a VMWare hypervisor v7.0 with Intel Xeon E5-4620 V4 processor (Broadwell), processor, at the Acumen Security office located at 2400 Research Blvd Suite #395, Rockville, MD 20850. Testing occurred from June 2023 to May 2024.

5 Detailed TSS and Guidance Evaluation Activities

5.1 TSS and Guidance Activities (Cryptographic Support)

5.1.1 FCS_CKM_EXT.1

5.1.1.1 FCS_CKM_EXT.1 TSS 1 [**TD0717]**

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FCS_CKM.1/AK** in the Security Target to determine if the application needs asymmetric key generation services. The evaluator examined the SFR section in the Security Target and determined that the application needs asymmetric key generation services. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.2 FCS_CKM.1/AK

5.1.2.1 FCS_CKM.1/AK TSS 1 **[TD0717]**

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FCS_CKM.1/AK** in the Security Target to verify that the TSS identifies the key sizes supported by the TOE, and if more than one scheme is specified, the usage for each scheme. Upon investigation, the evaluator found that the TSS states that "**The TOE shall generate a 2048-bit RSA public-private key pair and construct a Certificate Signing Request (CSR) with that key pair."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.2.2 FCS_CKM.1/AK Guidance 1 **[TD0717]**

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TLS Common Criteria Settings** in the [AGD] to verify that it 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. Upon investigation, the evaluator found that the [AGD] states that "**The TOE generates an RSA 2048-bit key pair and constructs a Certificate Signing Request (CSR) with the public key. The CSR is sent to the**

	Endpoint Security Server to be signed which constructs an X.509 certificate and returns it to the TOE." Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.3 FCS_CKM.2

5.1.3.1 FCS_CKM.2 TSS 1 **[TD0717]**

Objective	The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1/AK.
Evaluator Findings	The evaluator examined the **TOE Summary Specification, Table 12, FCS_CKM.2** in the Security Target to verify that the supported key establishment schemes correspond to the key generation schemes identified in section FCS_CKM.1.1/AK in the security target. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.3.2 FCS_CKM.2 TSS 2 [**TD0717]**

Objective	If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.
Evaluator Findings	The evaluator examined the **TOE Summary Specification, Table 12, FCS_CKM.2** in the Security Target and determined that only one scheme is supported. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.3.3 FCS_CKM.2 Guidance 1 **[TD0717]**

Objective	The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).
Evaluator Findings	The evaluator examined the section titled **TLS Common Criteria Settings** in the [AGD] to verify that it instructs the administrator how to configure the TOE to use the selected key establishment scheme(s). Upon investigation, the evaluator found that the AGD states that "**To set the cipher suite used to TLS_RSA_WITH_AES_128_CBC_SHA, using a text editor, add an advanced section with an mxs/tls/cipher key to the configuration file, exactly as shown below:** **"advanced": {"mxs/tls/cipher": "!aNULL:!eNULL:!ECDSA:AES128-SHA},** **This limits the key exchange algorithm to RSA and encryption to AES 128 CBC."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.4 FCS_COP.1/SKC

5.1.4.1 FCS_COP.1/SKC Guidance 1 **[TD0717]**

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TLS Common Criteria Settings** in the [AGD] to verify that any configuration that is required to be done to configure the functionality for the required modes and key sizes is present. Upon investigation, the evaluator found that the [AGD] states that "**To set the cipher suite used to TLS_RSA_WITH_AES_128_CBC_SHA, use a text editor, add an advanced section with an mxs/tls/cipher key to the configuration file, exactly as shown below:** **"advanced": {"mxs/tls/cipher": "!aNULL:!eNULL:!ECDSA:AES128-SHA"},** **This limits the key exchange algorithm to RSA and encryption to AES 128 CBC."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.5 FCS_COP.1/Hash

5.1.5.1 FCS_COP.1/Hash TSS 1 **[TD0717]**

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FCS_COP.1/Hash** in the Security Target to verify that the TSS documents the association of the hash function with other application cryptographic functions. Upon investigation, the evaluator found that the TSS states that "**The TOE implements cryptographic hashing using SHA-1 and SHA-256 with message digest sizes 160 and 256 bits respectively. Implementation is in accordance with FIPS Pub 180-4 "Secure Hash Standard."** **SHA-1 is only used in the provisioning of the TOE, not in the digital signature functions.** **SHA-256 is used in TLS session negotiation and with HMACs used to verify the integrity of TLS traffic. SHA-256 is also used in conjunction with RSA as part of the Trellix Endpoint Security (HX) Series Appliance X.509 certificate verification."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.6 FCS_HTTPS_EXT.1/Client

5.1.6.1 FCS_HTTPS_EXT.1.1/Client TSS 1

Objective	The evaluator shall examine the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FCS_HTTPS_EXT.1/Client** in the Security Target to verify that the TSS provides enough detail to explain how the implementation complies with RFC 2818. Upon investigation, the evaluator found that the TSS states that "**The TOE implements the HTTPS protocol according**

	to RFC 2818 by implementing all SHALL, MUST, and SHOULD statements and by not implementing any SHALL NOT, MUST NOT, or SHOULD NOT statements." Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.7 FCS_RBG_EXT.1

5.1.7.1 FCS_RBG_EXT.1 TSS 1

Objective	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.
Evaluator Findings	The evaluator examined the SFR section in the Security Target and determined that "use no DRBG functionality" is not selected. Based on these findings, this assurance activity is considered satisfied.
Verdict	N/A, the TOE implements its own DRBG functionality.

5.1.7.2 FCS_RBG_EXT.1 TSS 2

Objective	If implement DRBG functionality is selected, the evaluator shall ensure that additional FCS_RBG_EXT.2 elements are included in the ST.
Evaluator Findings	The evaluator examined the Security Target and verified that additional FCS_RBG_EXT.2 elements are included. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.7.3 FCS_RBG_EXT.1 TSS 3

Objective	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.
Evaluator Findings	The evaluator examined the SFR section in the Security Target and determined that "invoke platform-provided DRBG functionality" is not selected. Based on these findings, this assurance activity is considered satisfied.
Verdict	N/A, the TOE implements its own DRBG functionality.

5.1.8 FCS_RBG_EXT.2

5.1.8.1 FCS_RBG_EXT.2.2 TSS 1

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

Evaluator Findings	An Entropy Assessment report, which has been approved by NIAP, is provided.
Verdict	Pass

5.1.9 FCS_STO_EXT.1

5.1.9.1 FCS_STO_EXT.1 TSS 1

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FCS_STO_EXT.1** in the Security Target to verify that the TSS 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. Upon investigation, the evaluator found that the TSS states that "**The TOE stores digital certificates and private keys in the TOE's JSON structure stored in non-volatile memory. The database is encrypted by the TOE with AES and is not accessible to any external entity. The certificates are used to validate the HX server certificate. The RSA key pairs are used for encryption, decryption, and digital signatures during a TLS communication with only the HX server."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.10 FCS_TLS_EXT.1

5.1.10.1 FCS_TLS_EXT.1 Guidance 1

Objective	The evaluator shall ensure that the selections indicated in the ST are consistent with selections in the dependent components.
Evaluator Findings	The evaluator examined the section titled **'TLS Common Criteria Settings'** in the [AGD] to verify that the selections indicated in the ST are consistent with selections in the dependent components. Upon investigation, the evaluator found that the AGD states that "**The TOE only implements TLS as a sender."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.11 FCS_TLSC_EXT.1

5.1.11.1 FCS_TLSC_EXT.1.1 TSS 1 **[TD0442]**

Objective	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.

Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FCS_TLS_EXT.1** in the Security Target to verify that the TSS specifies the cipher suites supported and that the cipher suites specified include those listed for this component. Upon investigation, the evaluator found that the TSS states that "**The TOE only communicates with the Trellix Endpoint Security (HX) appliance using:** – **TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.11.2 FCS_TLSC_EXT.1.1 Guidance 1 **[TD0442]**

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TLS Common Criteria Settings** in the [AGD] and verified that it contains instructions on configuring the product so that TLS conforms to the description in the TSS. It provides instructions for establishing TLS protocols, certificate verification of the reference identifier and other configuration settings required to limit the ciphersuites. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.11.3 FCS_TLSC_EXT.1.2 TSS 1 **[TD0499]**

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FCS_TLSC_EXT.1.2 and FIA_X509_EXT.1** in the Security Target to verify 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 and whether IP addresses and wildcards are supported. Upon investigation, the evaluator found that the TSS states that "**X.509 certificates used for this connection are validated using the certificate path validation algorithm defined in RFC 5280. This includes performing a bit-by-bit verification of the reference identifier. As part of the x509 certificate validation process for the HX server during TLS connectivity establishment, the TOE verifies the reference identifier found in the SAN (Subject Alternative Name) or the CN (Common Name) field. The TOE supports exact matching and wildcard usage for both types of identifiers, with the condition that the wildcard is the entire left-most label. However, it does not support IP address as reference identifier or certificate pinning."**. **"If the TOE cannot establish a connection to verify the validity of a certificate, it will reject the certificate. It's important to note that the TOE does not support pinned certificates. While wildcards are supported, they only match in the left-most label and do not match with labels featuring an explicit prefix or suffix."**

	Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.11.4 FCS_TLSC_EXT.1.2 TSS 2 **[TD0499]**

Objective	The evaluator shall ensure that this description identifies whether and the manner in which certificate pinning is supported or used by the product.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FIA_X509_EXT.1** in the Security Target to verify that the TSS identifies whether and the manner in which certificate pinning is supported or used by the product. Upon investigation, the evaluator found that the TSS states that "**It's important to note that the TOE does not support pinned certificates**." Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.11.5 FCS_TLSC_EXT.1.2 Guidance 1 [**TD0499]**

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TLS Common Criteria Settings** in the [AGD] to verify that it includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS. Upon investigation, the evaluator found that the [AGD] states that "**When the TLS connection is being established, the agent verifies the reference identifier found in the SAN (Subject Alternative Name) or the CN (Common Name) field in case the SAN is not configured in the X509 presented certificate of the HX server. The TOE supports exact matching and wildcard usage for both types of identifiers, with the condition that the wildcard is the entire left-most label. However, it does not support IP address as reference identifier or certificate pinning.** **Add the following option to the configuration file to enforce verification of the identity of the Endpoint Security server: "fips": { "enabled": true, "hostnames": "host1,myuid.uid.crt" }, Where "host1,myuid.uid.crt" matches the CN/SAN of the certificate'."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.11.6 FCS_TLSC_EXT.1.3 TSS 1

Objective	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.
Evaluator Findings	The evaluator examined the SFR in the Security Target and determined that "except when override is authorized" is not selected. Based on these findings, this assurance activity is considered satisfied.

Verdict	N/A, the selection for authorizing override of invalid certificates is not made in the SFR.

5.1.12 FCS_TLSC_EXT.2

5.1.12.1 FCS_TLSC_EXT.2 TSS 1

Objective	The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication. The evaluator shall also ensure that the TSS describes any factors beyond configuration that are necessary in order for the client to engage in mutual authentication using X.509v3 certificates.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FCS_TLSC_EXT.2** in the Security Target to verify that the TSS includes the use of client-side certificates for TLS mutual authentication and any factors beyond configuration that are necessary in order for the client to engage in mutual authentication using X.509v3 certificates. Upon investigation, the evaluator found that the TSS states that "**In support of secure communication with external entities, the TOE implements the TLS protocol using mutual authentication mechanism. TLS is used to facilitate communication with the Trellix Endpoint Security (HX) Series Appliances."** **Also, in the same section, the TSS states that "All other forms of communication, except for fast polling, inherently utilize TLS encryption, thereby ensuring mutual authentication. No configuration is required on the TOE to enable it to participate in mutual authentication during TLS communication. The TOE will automatically transmit its x509 certificate when requested by the HX server during TLS communication. If the HX server does not send the certificate message request, the client will not transmit its certificate."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.1.12.2 FCS_TLSC_EXT.2 Guidance 1

Objective	The evaluator shall ensure that the AGD guidance includes any instructions necessary to configure the TOE to perform mutual authentication. The evaluator also shall verify that the AGD guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.
Evaluator Findings	The evaluator examined the section titled **TLS Common Criteria Settings** in the [AGD] and verified that includes instructions necessary to configure the TOE to perform mutual authentication and for configuring the client-side certificates. Upon investigation, the evaluator found that the [AGD] states that "**In support of secure communication with the endpoint security server, the agent implements the TLS protocol using mutual authentication mechanism. No configuration is required on the TOE to enable it to participate in mutual authentication during TLS communication. The TOE will automatically transmit its x509 certificate when requested by the HX server during TLS communication. If the HX server does not send the certificate message request, the client will not transmit its certificate."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.2 TSS and Guidance Activities (User Data Protection)

5.2.1 FDP_DAR_EXT.1

5.2.1.1 FDP_DAR_EXT.1 TSS 1

Objective	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 activities cover all of the sensitive data identified in the TSS.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FDP_DAR_EXT.1** in the Security Target to verify that the TSS describes the sensitive data processed by the application. Upon investigation, the evaluator found that the TSS states that: "**The only sensitive information stored by the Target of Evaluation (TOE) is its RSA cryptographic private key. Additionally, the TOE retains other non-sensitive information, including security policies, its x509 certificate, the TOE identity, and the associated Trellix Endpoint Security (HX) Series appliance identity**." The evaluator verified that the stated sensitive data is covered by the results obtained from the test assurance activities. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.2.1.2 FDP_DAR_EXT.1 TSS 2

Objective	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 .
Evaluator Findings	The evaluator examined the SFR in the Security Target and determined that "not store any sensitive data" is not selected. Based on these findings, this assurance activity is considered satisfied.
Verdict	N/A, the SFR in ST does not select "not store any sensitive data".

5.2.2 FDP_DEC_EXT.1

5.2.2.1 FDP_DEC_EXT.1.1 Guidance 1

Objective	The evaluator shall perform the platform-specific actions 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.
Evaluator Findings	The evaluator examined the section titled **Network Connectivity** in the [AGD] to verify that the stated hardware access is consistent with the SFR selections. Upon investigation, the evaluator found that the [AGD] states that "**The TOE requires access to the network to send and receive information from the Endpoint Security server:"** The evaluator also examined the section titled **Network Connectivity** in the [AGD] to verify that the stated hardware access is consistent with the results obtained from the test assurance activities.

	Upon investigation, the evaluator found that the hardware access information is consistent. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.2.2.2 FDP_DEC_EXT.1.1 Guidance 2

Objective	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.
Evaluator Findings	The evaluator examined the section titled **Network Connectivity** in the [AGD] to identify, for each resource which it accesses, the justification as to why access is required. Upon investigation, the evaluator found that the [AGD] states that "**The TOE requires access to the network to send and receive information from the Endpoint Security server:** • **The agent receives requests and updated security information from the Endpoint Security server.** • **The agent sends requested data and security information to the Endpoint Security server. "** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.2.2.3 FDP_DEC_EXT.1.2 Guidance 1

Objective	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.
Evaluator Findings	The evaluator examined the section titled **System Audit Examination** in the [AGD] to verify the sensitive information repositories. Upon investigation, the evaluator found that the [AGD] states that "**The TOE accesses system RAM, Filesystem and log files on the host machine while it is collecting information. As requested, some of this data may be transferred to the Endpoint Security server for further examination."** The evaluator also examined the section titled **System Audit Examination** in the [AGD] to verify that the stated repository access is consistent with the results obtained from the test assurance activities. Upon investigation, the evaluator found that the repository access information is consistent. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.2.2.4 FDP_DEC_EXT.1.2 Guidance 2

Objective	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.

Evaluator Findings	The evaluator examined the section titled **System Audit Examination** in the [AGD] to identify, for each sensitive information repository which it accesses, the justification as to why access is required. Upon investigation, the evaluator found that the [AGD] states that "**The agent monitors its host for host status and alert matches and reports this information to the server. Depending on acquisition parameters, the agent accesses the host file system to gather and deliver data for file and triage collection. It accesses RAM to gather the host full memory and system process memory for examination. It also accesses log files for host events.**" Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.3 TSS and Guidance Activities (Identification and Authentication)

5.3.1 FIA_X509_EXT.1

5.3.1.1 FIA_X509_EXT.1.1 TSS 1

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12_FIA_X509_EXT.1** in the Security Target to verify that the TSS describes where the check of validity of the certificates takes place and the certificate path validation algorithm. Upon investigation, the evaluator found that the TSS states that "**The TOE utilizes X.509v3 certificates, as defined in RFC 5280, to facilitate authentication for TLS connections. The validation of X.509 certificates follow the certificate path validation algorithm specified in RFC 5280, which encompasses the following checks:** • **Verification of the public key algorithm and parameters.** • **Validation of the current date and time against the certificate's validity period is performed for all TLS connections, with the exception of the initial time synchronization between the TOE and the HX server.** • **Verification of revocation status.** • **Matching the issuer name of certificate X with the subject name of certificate X+1.** • **Verification of name constraints.** • **Validation of policy Object Identifiers (OIDs).** • **Checking policy constraints, ensuring issuers possess CA signing capabilities.** • **Verification of the path length.** • **Processing of critical extensions."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.3.2 FIA_X509_EXT.2

5.3.2.1 FIA_X509_EXT.2.1 TSS 1

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FIA_X509_EXT.2** in the Security Target to verify that the TSS 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. Upon investigation, the evaluator found that the TSS states that **"The TOE's certificate will be signed by the HX server during the provisioning phase. Throughout operation, the TOE exclusively communicates with the HX server within a closed environment. Only one certificate is assigned to the TOE for its own use, meaning it will present only this certificate in cases where validation by the HX server is required."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.3.2.2 FIA_X509_EXT.2.1 TSS 2

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FIA_X509_EXT.2** in the Security Target to verify that the TSS 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. Upon investigation, the evaluator found that the TSS states that "**In cases where the PKI service is inactive, and the CRL is inaccessible, the TOE relies on the last known state of the HX certificate."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.4 TSS and Guidance Activities (Security Management)

5.4.1 FMT_CFG_EXT.1

5.4.1.1 FMT_CFG_EXT.1.1 TSS 1

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

Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FMT_CFG_EXT.1** in the Security Target to determine if the application requires any type of credentials and if the application installs with default credentials. Upon investigation, the evaluator found that the TSS states that "**The TOE does not necessitate any form of credentials for communication with the HX server, except for authentication via an x509 certificate. The Trellix Endpoint Security (HX) Server provides a TOE's certificate, in the agent configuration file that is included in agent download packages. Changing any of x509 certificates inputs in the initial TOE's configuration file will result in the failure to install the TOE. The certificates included in the TOE's configuration file are an x509 Certificate authority certificate in addition to the TOE's certificate."** The evaluator observed that the application does not install with default credentials. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.4.2 FMT_MEC_EXT.1

5.4.2.1 FMT_MEC_EXT.1 TSS 1

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FMT_MEC_EXT.1** in the Security Target to verify that the TSS identifies 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. Upon investigation, the evaluator found that the TSS states that "**After installing the agent software, you can configure specific settings stored in the agent_config.json file. Trellix advises against manually altering many settings in the agent_config.json file. Some settings in this file should only be modified with guidance from your Trellix support representative, typically for troubleshooting purposes. The TOE's guidance document outlines common settings in the agent_config.json file and explains whether and how they can be modified. It also provides instructions for editing the agent_config.json file and ensuring its validity after any changes. The TOE stores the settings configured during installation in the C:\ProgramData\FireEye\xagt directory.** Settings under the 'process' section and 'FIPS' section are among those that were set in the evaluated configuration.**"**. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.4.2.2 FMT_MEC_EXT.1 TSS 2

Objective	**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.

Evaluator Findings	The evaluator examined the SFR in the Security Target and verified that "implement functionality to encrypt and store configuration options as defined by FDP_PRT_EXT.1 in the PP-Module for File Encryption" **is not selected**. Based on these findings, this assurance activity is considered satisfied.
Verdict	N/A, because the option "implement functionality to encrypt and store configuration options as defined by FDP_PRT_EXT.1 in the PP-Module for File Encryption" is not selected in the SFR.

5.4.3 FMT_SMF.1

5.4.3.1 FMT_SMF.1 Guidance 1

Objective	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.
Evaluator Findings	The evaluator examined the [AGD] to 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. Upon investigation, the evaluator found that the [AGD], section: **Other Common Criteria Settings**, states that "**The Common Criteria configuration mandates that Windows Defender Exploit Guard protection be enabled. To achieve this, please ensure that the agent configuration under the 'process' node is configured as follows:** **"process": {** **"priority": "idle",** **"cpu_limit": 100,** **"deny_local_admin_stop": false,** **"detect_unsigned_image_loads": true,** **"file_protection_enabled": false,** **"protection_enabled": false,** **"strict_certificate_validation": true** **},** **Note: Some administrative parameters in this system are modified through editing JSON configuration files. The content of the Json file will be validated during the next restart of the TOE's services, therefore wrong syntax or modification to protected fields will not be imported to the TOE. Adjustments of parameters through JSON file editing should be ideally under the guidance of vendor support representatives. Administrators should also double check both syntax and semantics of any change before saving the JSON file and directing the system to ingest the change**." Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.5 TSS and Guidance Activities (Privacy)

5.5.1 FPR_ANO_EXT.1

5.5.1.1 FPR_ANO_EXT.1 TSS 1

Objective	The evaluator shall inspect the TSS documentation to identify functionality in the application where PII can be transmitted.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FPR_ANO_EXT.1** in the Security Target to verify that the TSS identifies functionality in the application where PII can be transmitted. Upon investigation, the evaluator found that the TSS states that "**The TOE does not transmit PII over the network.**" Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.6 TSS and Guidance Activities (Protection of the TSF)

5.6.1 FPT_AEX_EXT.1

5.6.1.1 FPT_AEX_EXT.1.1 TSS 1 **[TD0798]**

Objective	The evaluator shall ensure that the TSS describes the compiler flags used to enable ASLR when the application is compiled. If any explicitly-mapped exceptions are claimed, the evaluator shall check that the TSS identifies these exceptions, describes the static memory mapping that is used, and provides justification for why static memory mapping is appropriate in this case.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FPT_AEX_EXT.1** in the Security Target to verify that the TSS describes the compiler flags used to enable ASLR when the application is compiled. Upon investigation, the evaluator found that the TSS states that "**During compilation the TOE is built with several flags enabled that check for engineering flaws. The flags used (or not used) are the following:** – **The TOE is built with the /GS flag enabled. This reduces the possibilities of stack-based buffer overflows in the product.** – **The compiler enables ASLR by default.** **The TOE is not compiled with the /DYNAMICBASE:NO which would disable ASLR."** additionally, the evaluator observed that the SFR:FPT_AEX_EXT.1 in the ST, does not claim any explicitly-mapped exceptions. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.6.1.2 FPT_AEX_EXT.1.5 TSS 1 [**TD0815]**

Objective	(Conditional: The PE or ELF automated tests fail) The evaluator shall ensure that the TSS describes the stack-based buffer overflow compiler flags.

Evaluator Findings	The PE or RLF automated tests did not fail (see section 6.1.51), therefore this assurance activity is not applicable. Based on these findings, this assurance activity is considered satisfied.
Verdict	N/A. The PE or RLF automated tests did not fail.

5.6.2 FPT_API_EXT.1

5.6.2.1 FPT_API_EXT.1 TSS 1

Objective	The evaluator shall verify that the TSS lists the platform APIs used in the application.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FPT_API_EXT.1** in the Security Target to verify that the TSS lists the platform APIs used in the application. Upon investigation, the evaluator found that the TSS states that "**The TOE leverages the following platform provided Application Programing Interfaces (APIs):** – **CryptAcquireContextW** – **CryptGenRandom** – **CryptReleaseContext** – **CryptProtectData** - **CryptUnprotectData"** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.6.3 FPT_IDV_EXT.1

5.6.3.1 FPT_IDV_EXT.1 TSS 1

Objective	If "other version information" is selected the evaluator shall verify that the TSS contains an explanation of the versioning methodology.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FPT_IDV_EXT.1** in the Security Target to verify that the TSS contains an explanation of the versioning methodology. Upon investigation, the evaluator found that the TSS states that "**The TOE is distributed as a host platform specific package file providing a consistent and reliable versioning. The xAgent deployed with the .swidtag versioning file in the installation directory."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.6.4 FPT_TUD_EXT.1

5.6.4.1 FPT_TUD_EXT.1.1 Guidance 1

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

Evaluator Findings	The evaluator examined the section titled **Updating the TOE** in the [AGD] to verify that it includes a description of how updates are performed. Upon investigation, the evaluator found that the [AGD] describes in detail the methods to update the TOE. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.6.4.2 FPT_TUD_EXT.1.2 Guidance 1

Objective	The evaluator shall verify guidance includes a description of how to query the current version of the application.
Evaluator Findings	The evaluator examined the section titled **Verifying the TOE Version** in the [AGD] to verify that it includes a description of how to query the current version of the application. Upon investigation, the evaluator found that the above-mentioned section states that "**You can determine the agent version in two ways.** **1. Run the xagt exe –version command on a command line.** **2. Right-click on the agent executable file (xagt.exe), wherever it is installed. Select Properties on the drop-down menu and click the Details tab of the Properties dialog box."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.6.4.3 FPT_TUD_EXT.1.4 TSS 1

Objective	The evaluator shall verify that the TSS identifies how updates to the application are signed by an authorized source. The definition of an authorized source must be contained in the TSS. The evaluator shall also ensure that the TSS (or the operational guidance) describes how candidate updates are obtained.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FPT_TUD_EXT.1** in the Security Target to verify that the TSS identifies how updates to the application are signed by an authorized source. Upon investigation, the evaluator found that the TSS states that "**TOE updates are signed using digital certificates. The MSI packages are signed using certificates from a public trust chain which leads to DigiCert. Some components of the installation package (for instance, RemediationWSC), are signed using certificates with a public trust chain which leads to Sectigo**." The evaluator also examined the section titled **TOE Summary Specification, Table 12, FPT_TUD_EXT.1** in the Security Target to verify that the TSS (or the operational guidance) describes how candidate updates are obtained. Upon investigation, the evaluator found that the TSS states that "**The user can use the platform provided web browser to query the HX series appliance to determine if an update is available."** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.6.4.4 FPT_TUD_EXT.1.5 TSS 1

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FPT_TUD_EXT.1** in the Security Target to verify that the TSS identifies how the application is distributed. Upon investigation, the evaluator found that the TSS states that "**The TOE, initial installation as well as updates, is distributed as an operating system specific MSI package file.**" Upon investigation, the evaluator found that this SFR in ST, does not select the option "with the platform". Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.6.5 FPT_TUD_EXT.2

5.6.5.1 FPT_TUD_EXT.2.3 TSS 1

Objective	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.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, FPT_TUD_EXT.2** in the Security Target to verify that the TSS identifies how the application installation package is signed by an authorized source. Upon investigation, the evaluator found that the TSS states that "**The updates to the TOE are distributed as MSI package files. The MSI package files are signed using certificates with a public trust chain which leads to DigiCert. Some components of the installation package (for instance, RemediationWSC), are signed using certificates with a public trust chain which leads to Sectigo.**" Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

5.7 TSS and Guidance Activities (Trusted Path/Channels)

5.7.1 FTP_DIT_EXT.1

5.7.1.1 FTP_DIT_EXT.1 TSS 1

Objective	For platform-provided functionality, the evaluator shall verify the TSS contains the calls to the platform that TOE is leveraging to invoke the functionality.
Evaluator Findings	The evaluator examined the SFR section in the Security Target and determined that platform-provided functionality is not selected. Based on these findings, this assurance activity is considered satisfied.

Verdict	Not applicable, as all cryptographic functionality, including the TLS protocol, is provided by the OpenSSL library, which is included within the TOE boundary.

6 Detailed Testing Evaluation Activities

6.1 APP_V1.4

6.1.1 FCS_CKM.1/AK Test/CAVP 1

Item	Data
Test Assurance Activity	If the application "implements asymmetric key generation," then the following test activities shall be carried out. Evaluation Activity Note: The following tests may require the developer to provide access to a developer environment that provides the evaluator with tools that are typically available to endusers 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, • $\lvert p-q \rvert > 2^{nlen/2 - 100}$, • $p \geq 2^{nlen/2 -1/2}$, • $q \geq 2^{nlen/2 -1/2}$, • $2^{(nlen/2)} < d < LCM(p-1,q-1)$, • $e \cdot d = 1 \bmod LCM(p-1,q-1)$. **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, P384 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 $1 \leq x \leq q-1$ • len(q) + 64 bit output of RBG, followed by a mod q-1 operation where $1 \leq x \leq q-1$. 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 • $g \neq 0,1$ • q divides p-1 • $g^q \bmod p = 1$ • $g^x \bmod p = y$ 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.
Test Steps	**Key Generation for FIPS PUB 186-4 RSA Schemes** The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A5228 for FIPS186-4 RSA key generation using the key size 2048. This certificate provides assurance that the TSF performs these functions as required. **Key Generation for Elliptic Curve Cryptography (ECC)**

	ECC tests are not applicable as the TOE does not use or claim ECC key generation. **Key Generation for Finite-Field Cryptography (FFC)** FFC tests are not applicable as the TOE does not use or claim FCC key generation. **Diffie-Hellman Group 14 and FFC Schemes using "safe-prime" groups** DH tests are not applicable as the TOE does not use or claim DH key generation.
Pass/Fail with Explanation	**Key Generation for FIPS PUB 186-4 RSA Schemes** Pass **Key Generation for Elliptic Curve Cryptography (ECC)** N/A **Key Generation for Finite-Field Cryptography (FFC)** N/A because FFC tests are not applicable as the TOE does not use or claim FCC key generation. **Diffie-Hellman Group 14 and FFC Schemes using "safe-prime" groups.** N/A because DH tests are not applicable as the TOE does not use or claim DH key generation.

6.1.2 FCS_CKM.2 Test/CAVP 1

Item	Data
Test Assurance Activity	**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 rolekey 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.
Test Steps	**SP800-56A Key Establishment Schemes** SP800-56 tests are not applicable as the TOE does not use or claim SP800-56A key establishment schemes. **SP800-56B Key Establishment Schemes** Vendor affirmation to SP 800-56B, more detailed are in the TSS section of the SFR: FCS_CKM.2 **RSA-based key establishment** RSA-based tests are not applicable as the TOE does not use or claim RSAES-PKCS1-v1_5 key establishment schemes. **Diffie-Hellman Group 14.** Diffie-Hellman tests are not applicable as the TOE does not use or claim Diffie-Helman Group 14 key establishment schemes. **FFC Schemes using "safe-prime" groups.** FCC Schemes tests are not applicable as the TOE does not use or claim safe-prime groups key establishment schemes.

Pass/Fail with Explanation	**SP800-56A Key Establishment Schemes** N/A because the TOE does not use or claim SP800-56A key establishment schemes. **SP800-56B Key Establishment Schemes** Pass **RSA-based key establishment** N/A because the TOE does not use or claim RSAES-PKCS1-v1_5 key establishment schemes. **Diffie-Hellman Group 14** N/A because the TOE does not use or claim Diffie-Helman Group 14 key establishment schemes. **FFC Schemes using "safe-prime" groups** N/A because the TOE does not use or claim safe-prime groups key establishment schemes.

6.1.3 FCS_COP.1/SKC Test/CAVP 1

Item	Data
Test Assurance Activity	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

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 AESCCM, 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** **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 lessthan 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.
Test Steps	**AES-CBC Known Answer Tests** The evaluator examined the ST and found that in Section “Cryptographic Support” that the TOE was awarded the CAVP certificate #A5228 for AES-CBC (NIST SP800-38A) using key sizes 128 and 256 for encryption and decryption. This certificate provides assurance that the TSF performs these functions as required. **AES-CBC Multi-Block Message Test** The evaluator examined the ST and found that in Section “Cryptographic Support” that the TOE was awarded the CAVP certificate #A5228 for AES-CBC (NIST SP800-38A) using key sizes 128 and 256 for encryption and decryption. This certificate provides assurance that the TSF performs these functions as required.

	AES-GCM Monte Carlo Tests This is not applicable as the TOE does not claim or use AES in XTS mode. **AES-XTS Tests** This is not applicable as the TOE does not claim or use AES in XTS mode. **AES-CCM Tests** This is not applicable as the TOE does not claim or use AES in CCM mode. **AES-CTR Tests** This is not applicable as the TOE does not claim to use AES in CTR mode.
Pass/Fail with Explanation	**AES-CBC Known Answer Tests** Pass **AES-CBC Multi-Block Message Test** Pass **AES-GCM Monte Carlo Tests** N/A because the TOE does not claim AES in CBC mode. **AES-XTS Tests** N/A because the TOE does not claim AES in XTS mode. **AES-CCM Tests** N/A because the TOE does not claim AES in CCM mode. **AES-CTR Tests** N/A because the TOE does not claim AES in CTR mode.

6.1.4 FCS_COP.1/Hash Test/CAVP 1

Item	Data
Test Assurance Activity	The TSF hashing functions can be implemented in one of two modes. The first mode is the byteoriented 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.
Test Steps	The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A5228 for SHA-1 (FIPS Pub 180-4) and SHA2-256 (FIPS Pub 180-4). This certificate provides assurance that the TSF performs these functions as required.
Pass/Fail with Explanation	Pass.

6.1.5 FCS_COP.1/Sig Test/CAVP 1

Item	Data
Test Assurance Activity	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., P256, 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.
Test Steps	**ECDSA Algorithm Tests** ECDSA tests are not applicable as the TOE does not use or claim ECDSA algorithm support. **RSA Signature Algorithm Tests** The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A5228 for RSA signature generation and signature verification (FIPS Pub 186-4) using 2048-bit RSA keys. This certificate provides assurance that the TSF performs these functions as required.
Pass/Fail with Explanation	**ECDSA Algorithm Tests** N/A because ECDSA is not claimed. **RSA Signature Algorithm Tests** Pass

6.1.6 FCS_COP.1/KeyedHash Test/CAVP 1

Item	Data
Test Assurance Activity	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.
Test Steps	The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A5228 for HMAC-SHA2-256 (FIPS Pub 198-1) and HMAC-SHA-1 (FIPS Pub 198-1). This certificate provides assurance that the TSF performs these functions as required.
Pass/Fail with Explanation	Pass.

6.1.7 FCS_RBG_EXT.2.1 Test/CAVP 1

Item	Data
Test Assurance Activity	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.
Test Steps	**Implementations Conforming to FIPS 140-2 Annex C** This test is not applicable because the TOE does not claim conformance to FIPS 140-2 Annex C. **Implementations Conforming to NIST Special Publication 800-90A**

	The evaluator examined the ST and found that in Section “Cryptographic Support” that the TOE was awarded the CAVP certificate #A5228 for Counter DRBG (NISP SP 800-90A) (AES-256). This certificate provides assurance that the TSF performs these functions as required.
Pass/Fail with Explanation	**Implementations Conforming to FIPS 140-2 Annex C** N/A because the TOE does not claim conformance to FIPS 140-2 Annex C. **Implementations Conforming to NIST Special Publication 800-90A** Pass

6.1.8 FCS_HTTPS_EXT.1.1/Client Test #1

Item	Data
Test Assurance Activity	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.
Test Steps	• Initiate an HTTPS connection to the HX Server from the TOE by running the application. • Observe the network traffic on the packet capture and verify that it is protected by TLS or HTTPS.
Expected Test Results	• Successful HTTPS connection to the HX server. • Packet capture showing network traffic is protected by TLS or HTTPS.
Pass/Fail with Explanation	PASS. HTTPS connection to the HX server is protected by TLS. This meets the testing requirements.

6.1.9 FCS_HTTPS_EXT.1.2/Client Test #1

Other tests are performed in conjunction with the TLS package.

6.1.10 FCS_HTTPS_EXT.1.3/Client Test #1

Item	Data
Test Assurance Activity	Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1, and the evaluator shall perform the following test: 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**" is selected in the SFR, then the evaluator shall also determine that the user is notified 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.
Test Steps	<u>Valid certificate chain</u> • Create a valid certificate chain. • Ensure that the TOE contains the valid CA certificate in its config file. • Attempt a TLS connection from the TOE to the TLS Server using a certificate with a valid certification path and show the connection being successful. • Verify the successful connection with the packet capture. <u>Invalid certificate chain</u> • Create an invalid certificate chain.

	• Attempt a TLS connection from the TOE to the TLS Server using a certificate with an invalid certification path and show the connection being unsuccessful. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	• Packet capture showing successful TLS connection when a valid certificate chain is presented. • Packet capture showing unsuccessful TLS connection when an invalid certificate chain is presented.
Pass/Fail with Explanation	PASS. The function succeeds when a valid certificate chain is loaded and leads to validation failure when a certificate with an invalid certification path is presented. This meets the testing requirements.

6.1.11 FCS_HTTPS_EXT.2.1 Test #1

Item	Data
Test Assurance Activity	Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1, and the evaluator shall perform the following test: The evaluator shall demonstrate that using a certificate without a valid certification path results in the selected action in the SFR. 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.
Test Steps	<u>Valid certificate chain</u> • Create a valid certificate chain. • Ensure that the TOE contains the valid CA certificate in its config file. • Attempt a TLS connection from the TOE to the TLS Server using a certificate with a valid certification path and show the connection being successful. • Verify the successful connection with the packet capture. <u>Invalid certificate chain</u> • Create an invalid certificate chain. • Attempt a TLS connection from the TOE to the TLS Server using a certificate with an invalid certification path and show the connection being unsuccessful. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	• Packet capture showing successful TLS connection when a valid certificate chain is presented. • Packet capture showing unsuccessful TLS connection when an invalid certificate chain is presented.
Pass/Fail with Explanation	PASS. The function succeeds when a valid certificate chain is loaded and leads to validation failure when a certificate with an invalid certification path is presented. This meets the testing requirements.

6.1.12 FCS_RBG_EXT.1.1

Item	Data
Test Assurance Activity	If **invoke platform-provided DRBG functionality** is selected, the following tests shall be performed The evaluator shall decompile the application binary using a decompiler suitable for the application (TOE). The evaluator shall search the output of the decompiler to determine

	that, for each API listed in the TSS, that API appears in the output. If the representation of the API does not correspond directly to the strings in the following list, the evaluator shall provide a mapping from the decompiled text to its corresponding API, with a description of why the API text does not directly correspond to the decompiled text and justification that the decompiled text corresponds to the associated API. The following are the per-platform list of acceptable APIs: **Platforms:Microsoft 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.
Pass/Fail with Explanation	N/A, the ST does not select **"invoke platform-provided DRBG functionality"**.

6.1.13 FCS_RBG_EXT.2.2

Item	Data
Test Assurance Activity	In the future, specific statistical testing (in line with NIST SP 800-90B) will be required to verify the entropy estimates.
Test Steps	N/A
Pass/Fail with Explanation	N/A

6.1.14 FCS_STO_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	For all credentials for which **the application implements functionality**, the evaluator shall verify credentials are encrypted according to FCS_COP.1/SKC or conditioned according to FCS_CKM.1.1/AK and FCS_CKM.1/PBKDF.
Test Steps	• Locate the database file of the TOE. • Open it with text editor and verify that the file contents are encrypted. • Attempt to open the database file with SQLCipher and ensure it is unsuccessful without the encryption key.
Expected Test Results	File contents and credentials stored in the main.db should be encrypted.
Pass/Fail with Explanation	PASS. Certificates and credentials in the main database file of the TOE are encrypted. This meets the testing requirements.

6.1.15 FCS_STO_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	For all credentials for which the application **invokes platform-provided functionality**, the evaluator shall perform the following actions which vary per platform. **Platforms:Microsoft Windows...** The evaluator shall verify that all certificates are stored in the Windows Certificate Store. The evaluator shall verify that other credentials, like passwords, are stored in the

	Windows Credential Manager or stored using the Data Protection API (DPAPI). For Windows Universal Applications, the evaluator shall verify that the application is using the ProtectData class and storing credentials in IsolatedStorage.
Pass/Fail with Explanation	N/A, the ST does not select **"invokes platform-provided functionality"**.

6.1.16 FDP_DAR_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	Evaluation activities (after the identification of the sensitive data) are to be performed on all sensitive data listed that are not covered by FCS_STO_EXT.1. If "implement functionality to encrypt sensitive data as defined in the PP-Module for File Encryption" or "protect sensitive data in accordance with FCS_STO_EXT.1" is selected, the evaluator shall inventory the filesystem locations where the application may write data. The evaluator shall run the application and attempt to store sensitive data. The evaluator shall then inspect those areas of the filesystem to note where data was stored (if any), and determine whether it has been encrypted. **TD0756 has been applied.**
Pass/Fail with Explanation	PASS. All the sensitive data listed is covered as part of FCS_STO_EXT.1. There is no sensitive data listed that is not covered as part of FCS_STO_EXT.1. This meets the testing requirements.

6.1.17 FDP_DAR_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	Evaluation activities (after the identification of the sensitive data) are to be performed on all sensitive data listed that are not covered by FCS_STO_EXT.1. 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. **Platforms:Microsoft 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.
Pass/Fail with Explanation	N/A, the ST does not select **"leverage platform-provided functionality"**.

6.1.18 FDP_DEC_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	**Platforms:Microsoft Windows...** For Windows Universal Applications the evaluator shall check the AppxManifest.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.

	TD0822 has been applied.

Test Steps	• List of the required hardware capabilities as per the ST.
	  ○ SFR section in ST.
	  ○ TSS section in ST.
	• Check the AGD and verify that it lists the required hardware resources.

Expected Test Results	• Screenshot evidence of the ST where it lists the hardware capabilities of the TOE. • Screenshot evidence of the AGD where it lists the hardware capabilities of the TOE.
Pass/Fail with Explanation	PASS. The TOE documentation lists the required hardware resources along with necessary justification. This meets the testing requirements.

6.1.19 FDP_DEC_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	**Platforms:Microsoft Windows...** For Windows Universal Applications the evaluator shall check the AppxManifest.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. **TD0822 has been applied.**

Test Steps	• List of the sensitive information repositories it accesses as per the ST.
	  ○ SFR section in ST.
	  ○ TSS section in ST.
	• Check the AGD and verify that it lists the sensitive information repositories it accesses.

Expected Test Results	• Screenshot evidence of the ST where it lists the sensitive information repositories the TOE accesses. • Screenshot evidence of the AGD where it lists the sensitive information repositories the TOE accesses.
Pass/Fail with Explanation	PASS. The TOE documentation lists the sensitive information repositories the TOE accesses. This meets the testing requirements.

6.1.20 FDP_NET_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	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 Steps	• Run the application. • Verify the network communications with the packet capture. • Verify that the network communications witnessed are documented in the TSS.
Expected Test Results	• Screenshot evidence of the packet captures to verify the network communications. • Screenshot evidence of the TSS mentioned with all the witnessed network communications.
Pass/Fail with Explanation	PASS. It has been verified that the network communications witnessed are documented in the TSS. This meets the testing requirements.

6.1.21 FDP_NET_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	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).
Test Steps	• Run the application. • On the HX server, choose the TOE Host machine under “Hosts>All Hosts” and request for information by selecting an option from the “Acquire” drop-down list. • Run a port scan using Netstat and verify the ports opened by the application to communicate with the HX server. • Verify the ports opened by the application have been captured in the ST.
Expected Test Results	• Netstat showing the ports opened by the application. • Screenshot evidence showing that the ports opened by the application are documented in the ST.
Pass/Fail with Explanation	PASS. The ports opened by the application for network communications are documented in the ST. This meets the testing requirements.

6.1.22 FDP_NET_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	**Platforms: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.
Pass/Fail with Explanation	N/A. TOE is evaluated on Windows Platform.

6.1.23 FIA_X509_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 1: The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function failing, for each of the following reasons, in turn: • by establishing a certificate path in which one of the issuing certificates is not a CA certificate, • by omitting the basicConstraints field in one of the issuing certificates, • by setting the basicConstraints field in an issuing certificate to have CA=False,

	• by omitting the CA signing bit of the key usage field in an issuing certificate, and • by setting the path length field of a valid CA field to a value strictly less than the certificate path. The evaluator shall then establish a valid certificate path consisting of valid CA certificates, and demonstrate that the function succeeds. The evaluator shall then remove trust in one of the CA certificates, and show that the function fails.
Test Steps	• Create a certificate chain using the XCA tool. <u>Establish a certificate path in which one of the issuing certificates is not a CA certificate:</u> • Transform similar certificate of the issuing certificate and make it to a non-CA certificate. • PEM data of the non-CA certificate. • Import the non-CA certificate to the agent_config file. • Import the server certificate to the Apache server in its required path for TLS connection. • Install the application with the updated config file to establish a TLS connection with the Server. • Verify the failure connection logs on the TOE. <u>By omitting the basicConstraints field in one of the issuing certificates:</u> • Transform similar certificate of the issuing certificate and make it to a certificate lacking the basicConstraints field. • PEM data of the certificate with no basicConstraints. • Import the certificate with no basicConstraints to the agent_config file. • Install the application with the updated config file to establish a TLS connection with the Server. • Verify the failure connection logs on the TOE. <u>By setting the basicConstraints field in an issuing certificate to have CA=False:</u> • Modify the valid certificate to have CA=False with the help of acumen x509-mod tool. • PEM data of the non-CA certificate. • Import the non-CA certificate to the agent_config file. • Install the application with the updated config file to establish a TLS connection with the Server. • Verify the failure connection logs on the TOE. <u>By omitting the CA signing bit of the key usage field in an issuing certificate:</u> • Transform similar certificate of the issuing certificate and omit the CA signing bit of the key usage field. • PEM data of the certificate with no CA signing. • Import the certificate with no CA signing to the agent_config file. • Install the application with the updated config file to establish a TLS connection with the Server. • Verify the failure TLS connection logs on the TOE. • Verify the rejection of TLS connection with the packet capture. <u>By setting the path length field of a valid CA field to a value strictly less than the certificate path:</u> • N/A. No intermediate CAs used for testing. <u>Valid certificate chain:</u> • Create a valid certificate chain with a valid CA certificate. • PEM data of the valid CA certificate.

	• Import the valid CA certificate to the agent_config file. • Install the application with the updated config file to establish a TLS connection with the Server. • Verify the successful TLS connection logs on the TOE. • Verify the successful TLS connection with the packet capture. Invalid certificate chain: • Create an invalid certificate chain. • Import the server certificate to the Apache server in its required path for TLS connection. • Install the application to establish a TLS connection with the Server. • Verify the failure TLS connection logs on the TOE. • Verify the unsuccessful TLS connection with the packet capture.
Expected Test Results	• TOE should not accept the certificate when one of the issuing certificates is not a CA certificate. • TOE should not accept the certificate when basicConstraints field in one of the issuing certificates is missing. • TOE should not accept the certificate when basicConstraints field in an issuing certificate have CA=False. • TOE rejecting the connection when CA signing bit of the key usage field in an issuing certificate is missing. • TOE rejecting the TLS connection with an invalid certificate path. • TOE successfully establishing the TLS connection with a valid certificate path. • Packet capture evidence demonstrating failed TLS connection when CA signing bit is missing in one of the issuing certificates. • Packet capture evidence demonstrating failed TLS connection with an invalid certification path. • Packet capture evidence demonstrating successful TLS connection with a valid certificate path.
Pass/Fail with Explanation	PASS. The evaluator observed that the TOE rejects a certificate without a valid certification path resulting in the communications channel not being established with the TLS server, for each of the following reasons: • One of the issuing certificates is not a CA certificate, • No basicConstraints field in one of the issuing certificates, • The basicConstraints field in an issuing certificate to have CA=False, • Missing the CA signing bit of the key usage field in an issuing certificate The test was not applicable for the following test step:" by setting the path length field of a valid CA field to a value strictly less than the certificate path.", because the TOE does not support intermediate CA. When all proper X509 conditions were met the evaluator observed the connection was successfully established. This meets the testing requirements.

6.1.24 FIA_X509_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1.

	The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 2: The evaluator shall demonstrate that validating **an expired certificate** results in the function failing.
Test Steps	Note: The application does not support chains of length four or greater. The application supports a maximum trust depth of two, hence a chain with no Intermediate CA will be created. • Create an expired server certificate. • Attempt a TLS connection from the TOE to the TLS Server using the expired server certificate and show the connection being unsuccessful. • Verify the unsuccessful connection using a collected packet capture.
Expected Test Results	• TOE rejecting the TLS connection when expired certificate is used. • Packet capture evidence showing unsuccessful TLS connection.
Pass/Fail with Explanation	PASS. The TOE does not accept an expired certificate and the TLS connection failed. This meets the testing requirements.

6.1.25 FIA_X509_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. **Test 3:** The evaluator shall test that the TOE can properly handle **revoked certificates** – conditional on whether CRL, OCSP, or OCSP Stapling or OCSP Multi-stapling is selected; if multiple methods are selected, then the following tests shall be performed for each method: o The evaluator shall test revocation of the node certificate. o 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.
Test Steps	Note: The application does not support chains of length four or greater. The application supports a maximum trust depth of two, hence a chain with no Intermediate CA will be created. Note: The OCSP option is not selected in ST. <u>Attempt a connection with a revoked Intermediate CA certificate:</u> • N/A, The TOE does not support the intermediate CA certificate. <u>Attempt a connection with a revoked server certificate:</u> • Create a certificate chain using the XCA tool and revoke the server certificate. • Generate the CRL using XCA tool. • Import the revoked server certificate to the Apache server in its required path for TLS connection. • Import the CRL to the Apache server in its required path for TLS connection. • Start the agent service to initiate a TLS connection with the server. • Verify that the TOE fetches the CRL file from the Apache server. • Ensure with packet capture that the TOE fetches the CRL. • Verify the unsuccessful connection with the packet capture. <u>Attempt a connection with a Valid server certificate:</u> • Create a certificate chain using the XCA tool with a valid server certificate. • Generate the CRL using XCA tool. • Import the server certificate to the Apache server in its required path for TLS connection. • Import the CRL to the Apache server in its required path for TLS connection. • Start the agent service to initiate a TLS connection with the server. • Verify that the TOE fetches the CRL file from the Apache server. • Ensure with packet capture that the TOE fetches the CRL. • Verify the successful TLS connection with the packet capture.
Expected Test Results	Note: The OCSP option is not selected in ST. • TOE rejecting the TLS connection when revoked certificate is presented. • TOE accepting the TLS connection when valid certificate is presented. • Packet capture evidence showing unsuccessful TLS connection when revoked certificate is presented. • Packet capture evidence showing successful TLS connection when valid certificates are presented.
Pass/Fail with Explanation	Pass. The OCSP option is not selected in the ST. For the CRL method, the TOE rejected the TLS connection for a revoked certificate and accepted the TLS connection with a valid certificate. This meets the testing requirements.

6.1.26 FIA_X509_EXT.1.1 Test #4

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules.

	If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 4: If any OCSP option is selected, the evaluator shall ensure the TSF has no other source of revocation information available and configure the OCSP server or use a man-in-the-middle tool to present an OCSP response signed by a certificate that does not have the OCSP signing purpose and which is the only source of revocation status information advertised by the CA issuing the certificate being validated. The evaluator shall verify that validation of the OCSP response fails and that the TOE treats the certificate being checked as invalid and rejects the connection. If CRL **is selected, the evaluator shall likewise configure the CA to be the only source of revocation status information**, and sign a CRL with a certificate that does not have the cRLsign key usage bit set. The evaluator shall verify that validation of the CRL fails and that the TOE treats the certificate being checked as invalid and rejects the connection. **TD0780 has been applied.**
Test Steps	Note: The application does not support chains of length four or greater. The application supports a maximum trust depth of two, hence a chain with no Intermediate CA will be created. Note: The OCSP option is not selected in ST. • Create a certificate chain with a CA certificate lacking the CRLsign bit. • Sign a CRL with the CA certificate lacking the CRLsign bit. • PEM data of the new CA certificate lacking the CRLsign bit. • Import the new CA certificate to the agent config file. • Import the server certificate to the Apache server in its required path for TLS connection. • Import the CRL file to the Apache server in its required path for TLS connection. • Install the application with the updated config file to establish a TLS connection with the Server. • Verify that the TOE fetches the CRL file from the Apache server. • Verify the failure TLS connection logs on the TOE. • Verify the CRL requests and rejection of TLS connection with the packet capture.
Expected Test Results	Note: The OCSP option is not selected in ST. • TOE rejecting the TLS connection when certificate does not have the CRLsign key usage bit set. • Packet capture evidence showing unsuccessful TLS connection.
Pass/Fail with Explanation	Pass. The OCSP option is not selected in the ST. The validation of the CRL failed with a CA certificate lacking the CRLsign bit, and the TLS connection was rejected. This meets the testing requirements.

6.1.27 FIA_X509_EXT.1.1 Test #5

Item	Data

Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 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 Steps	• Attempt a TLS connection using the acumen-tlsc tool and modify the first eight bytes of the certificate. • Verify the unsuccessful TLS connection on the packet capture.
Expected Test Results	• TOE rejecting the TLS connection when any byte in the first eight bytes of the certificate is modified. • Packet capture evidence showing unsuccessful TLS connection.
Pass/Fail with Explanation	PASS. The evaluator modified the first eight bytes of the certificate being presented by the server and ensured that the certificate fails to validate, and the TLS handshake fails. This meets the testing requirements.

6.1.28 FIA_X509_EXT.1.1 Test #6

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 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 Steps	• Attempt a TLS connection using the acumen-tlsc tool and modify any byte in the last byte of the certificate. • Verify the unsuccessful TLS connection on the packet capture.
Expected Test Results	• TOE rejecting the TLS connection when any byte in the last byte of the certificate is modified. • Packet capture evidence showing unsuccessful TLS connection.
Pass/Fail with Explanation	PASS. The evaluator modified the last byte of the certificate and demonstrated that the certificate fails to validate. This meets the testing requirements.

6.1.29 FIA_X509_EXT.1.1 Test #7

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 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 Steps	• Attempt a TLS connection using the acumen-tlsc tool and modify any byte in the public key of the certificate. • Verify the unsuccessful TLS connection on the packet capture.
Expected Test Results	• TOE rejecting the TLS connection when any byte in the public key of the certificate is modified. • Packet capture evidence showing unsuccessful TLS connection.
Pass/Fail with Explanation	PASS. The evaluator modified 8 bytes in the public key of the server certificate and demonstrated that the certificate fails to validate. This meets the testing requirements.

6.1.30 FIA_X509_EXT.1.1 Test #8

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 8: (**Conditional on support for EC certificates as indicated in FCS_COP.1/Sig**). The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.
Pass/Fail with Explanation	N/A, the ST does not select EC certificates in FCS_COP.1/Sig.

6.1.31 FIA_X509_EXT.1.1 Test #9

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. Test 9: (**Conditional on support for EC certificates as indicated in FCS_COP.1/Sig**). The evaluator shall replace the intermediate certificate in the certificate chain for Test 8a with a modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 8a, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.
Pass/Fail with Explanation	N/A, the ST does not select EC certificates in FCS_COP.1/Sig.

6.1.32 FIA_X509_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. 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.
Expected Test Results	The TOE rejects a TLS connection to a TLS server that presented a CA certificate that does not contain the basicConstraints extension.
Pass/Fail with Explanation	PASS. The TOE only supports a chain of two certificates. Section (i) of this test was performed in conjunction with FIA_X509_EXT.1.1 Test #1, bullet 2. The evaluator determined that the TOE rejected a TLS connection to a TLS server that presented a CA

	certificate that does not contain the basicConstraints extension. This meets the testing requirements.

6.1.33 FIA_X509_EXT.1.2 Test #2

Item	Data
Test Assurance Activity	The tests described must be performed in conjunction with the other certificate services evaluation activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. If the application supports chains of length four or greater, the evaluator shall create a chain of at least four certificates: - The node certificate to be tested, - Two Intermediate CAs, and - The self-signed Root CA. If the application supports a maximum trust depth of two, then a chain with no Intermediate CA should instead be created. 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
Pass/Fail with Explanation	Pass. Section (i) of this test was performed in conjunction with FIA_X509_EXT.1.1 Test #1 by setting the basicConstraints field in an issuing certificate to have CA=False and ensuring the connection fails. This meets the testing requirements.

6.1.34 FIA_X509_EXT.2.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each trusted channel: 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 Steps	1. Valid server certificate: <u>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.</u> • Create a certificate chain using the XCA tool with a valid server certificate. • Generate the CRL using XCA tool. • Import the server certificate to the Apache server in its required path for TLS connection. • Import the CRL to the Apache server in its required path for TLS connection. • Start the agent service to initiate a TLS connection with the server. • Verify that the TOE fetches the CRL file from the Apache server. • Ensure with packet capture that the TOE fetches the CRL.

	• Verify the successful TLS connection with the packet capture. Manipulating the environment so that the TOE is unable to verify the validity of the certificate. • Manipulate the environment by removing the CRL file from the Apache server. • Start the agent service to initiate a TLS connection with the server. • Verify that the TOE is unable to fetch the CRL file from the Apache server. • Ensure with packet capture that the TOE is unable to fetch the CRL. • Verify the successful TLS connection with the packet capture. 2. Revoked server certificate: Using a Revoked certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. • Revoke the server certificate using XCA tool. • Generate the CRL using XCA tool. • Import the revoked server certificate to the Apache server in its required path for TLS connection. • Import the CRL to the Apache server in its required path for TLS connection. • Start the agent service to initiate a TLS connection with the server. • Verify that the TOE fetches the CRL file from the Apache server. • Ensure with packet capture that the TOE fetches the CRL. • Verify the unsuccessful TLS connection with the packet capture. Manipulating the environment so that the TOE is unable to verify the validity of the certificate. • Manipulate the environment by removing the CRL file from the Apache server. • Start the agent service to initiate a TLS connection with the server. • Verify that the TOE is unable to fetch the CRL file from the Apache server. • Ensure with packet capture that the TOE is unable to fetch the CRL. • Verify the unsuccessful TLS connection with the packet capture.
Expected Test Results	• The TOE should accept the TLS connection when the certificate is valid and the CRL file is accessible. • The TOE should accept the TLS connection if the last known status of the certificate is valid when the CRL file is inaccessible. • The TOE should reject the TLS connection when the certificate is revoked and the CRL file is accessible. • The TOE should reject the TLS connection if the last known status of the certificate is revoked when the CRL file is inaccessible.
Pass/Fail with Explanation	Pass. If CRL is inaccessible, the TOE relies on the last known state of the certificate for TLS connection. Note: The only trusted channel claimed by the ST is the one between the TOE and the HX server. This meets the testing requirements.

6.1.35 FIA_X509_EXT.2.2 Test #2

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each trusted channel: 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.

Pass/Fail with Explanation	PASS. This test is performed in conjunction with FIA_X509_EXT.1.1 Test #2 and FIA_X509_EXT.1.1 Test #3 where certificate with Invalid path, Expired certificate and Revoked certificates resulted in connection failure. Note: The only trusted channel claimed by the ST is the one between the TOE and the HX server. This meets the testing requirements.

6.1.36 FMT_CFG_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	If the application uses any default credentials the evaluator shall run the following tests. **Test 1:** The evaluator shall install and run the application without generating or loading new credentials and verify that only the minimal application functionality required to set new credentials is available.
Pass/Fail with Explanation	N/A. The TOE does not support default credentials.

6.1.37 FMT_CFG_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	If the application uses any default credentials the evaluator shall run the following tests. **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.
Pass/Fail with Explanation	N/A. The TOE does not support default credentials.

6.1.38 FMT_CFG_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	If the application uses any default credentials the evaluator shall run the following tests. **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.
Pass/Fail with Explanation	N/A. The TOE does not support default credentials.

6.1.39 FMT_CFG_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall install and run the application. The evaluator shall inspect the filesystem of the platform (to the extent possible) for any files created by the application and ensure that their permissions are adequate to protect them. The method of doing so varies per platform. **Platforms:Microsoft Windows...** The evaluator shall run the SysInternals tools, Process Monitor and Access Check (or tools of equivalent capability, like icacls.exe) for Classic Desktop applications to verify that files written to disk during an application's installation have the correct file permissions, such that a standard user cannot modify the application or its data files. For Windows Universal Applications the evaluator shall consider the requirement met because of the AppContainer sandbox.

Test Steps	• Run AccessChk tool on the path "C:\Program Files(x86)\FireEye\xagt" and verify that the permissions are restricted to system administrators. • Run AccessChk tool on the path "C:\ProgramData\FireEye\xagt" and verify that the permissions are restricted to system administrators.
Expected Test Results	TOE application file permissions are restricted to system administrators.
Pass/Fail with Explanation	PASS. Files associated with the TOE are owned by the administrative user and therefore cannot be modified by a non-administrator. This meets the testing requirements.

6.1.40 FMT_MEC_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	If "**invoke the mechanisms recommended by the platform vendor for storing and setting configuration options**" is chosen, the method of testing varies per platform as follows: **Platforms: Microsoft Windows...** The evaluator shall determine and verify that Windows Universal Applications use either the Windows.Storage namespace, Windows.UI.ApplicationSettings namespace, or the IsolatedStorageSettings namespace for storing application specific settings. For .NET applications, the evaluator shall determine and verify that the application uses one of the locations listed in https://docs.microsoft.com/en-us/dotnet/framework/configure-apps/ for storing application specific settings. For Classic Desktop applications, the evaluator shall run the application while monitoring it with the SysInternals tool ProcMon and make changes to its configuration. The evaluator shall verify that ProcMon logs show corresponding changes to the the Windows Registry or C:\ProgramData\ directory. **TD0747 has been applied**
Test Steps	• Run the ProcMon tool and filter for the TOE (xagt.exe) and operation of "Write". • Modify the TOE configuration file, by editing a value in the agent config file and importing it to the agent executable. • Start the TOE service. • In ProcMon verify that changes made to the TOE configuration file are reflected in the Windows Registry or C:\ProgramData\FireEye\xagt.
Expected Test Results	TOE configuration file changes correspond to its application setting location in C:\ProgramData\FireEye\xagt.
Pass/Fail with Explanation	PASS. File changes made to the TOE config file are reflected in the Windows Registry or C:\ProgramData\FireEye\xagt. This meets the testing requirements.

6.1.41 FMT_MEC_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	N/A, the ST does not select *"implement functionality to encrypt and store configuration options as defined by FDP_PRT_EXT.1 in the PP-Module for File Encryption".*

6.1.42 FMT_SMF.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall test the application's ability to provide the management functions by configuring the application and testing each option selected from above. The evaluator is expected to test these functions in all the ways in which the ST and guidance documentation state the configuration can be managed.
Pass/Fail with Explanation	Pass. Refer to FPT_AEX_EXT.1.3 Test #1 for testing of management functions provided by the TOE.

6.1.43 FPR_ANO_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	N/A, the ST does not select "**require user approval before executing**".

6.1.44 FPT_AEX_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall perform either a static or dynamic analysis to determine that no memory mappings are placed at an explicit and consistent address except for any exceptions claimed in the SFR. For these exceptions, the evaluator shall verify that this analysis shows explicit mappings that are consistent with what is claimed in the TSS. The method of doing so varies per platform. For those platforms requiring the same application running on two different systems, the evaluator may alternatively use the same device. After collecting the first instance of mappings, the evaluator must uninstall the application, reboot the device, and reinstall the application to collect the second instance of mappings. **Platforms:Microsoft Windows...** The evaluator shall run the same application on two different Windows systems and run a tool that will list all memory mapped addresses for the application. The evaluator shall then verify the two different instances share no mapping locations. The Microsoft SysInternals tool, VMMap, could be used to view memory addresses of a running application. The evaluator shall use a tool such as Microsoft's BinScope Binary Analyzer to confirm that the application has ASLR enabled. **TD0798 has been applied.**
Test Steps	• Open VMMap on the first platform that the TOE is running and verify the memory mapped addresses of xagt.exe. • Open VMMap on the second platform that the TOE is running and verify the memory mapped addresses of xagt.exe. • Run BinScope Binary Analyzer scan the TOE process file xagt.exe located in C:\ProgramFiles(x86)\FireEye\xagt using the option "/Checks DBCheck" to verify that ASLR is enabled. • Verify that the DBCheck has passed.
Expected Test Results	• No memory mappings for the TOE process will be placed at an explicit and consistent address.

	• The BinScope DBCheck is successful.
Pass/Fail with Explanation	PASS. No memory mappings for the TOE are placed at an explicit and consistent address and the application has ASLR enabled. This meets the testing requirements.

6.1.45 FPT_AEX_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall verify that no memory mapping requests are made with write and execute permissions. The method of doing so varies per platform. **Platforms:Microsoft Windows...** The evaluator shall use a tool such as Microsoft's BinScope Binary Analyzer to confirm that the application passes the NXCheck. The evaluator may also ensure that the /NXCOMPAT flag was used during compilation to verify that DEP protections are enabled for the application.
Test Steps	• Using Microsoft BinScope Binary Analyzer scan the TOE process file xagt.exe located in C:\ProgramFiles(x86)\FireEye\xagt using the option "/Checks NXCheck". • Verify that the NXCheck has passed.
Expected Test Results	The BinScope NXCheck is successful.
Pass/Fail with Explanation	PASS. The TOE passes the NXCheck. This meets the testing requirements.

6.1.46 FPT_AEX_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure the platform in the ascribed manner and carry out one of the prescribed tests: **Platforms: Microsoft Windows...** If the OS platform supports Windows Defender Exploit Guard (Windows 10 version 1709 or later), then the evaluator shall ensure that the application can run successfully with Windows Defender Exploit Guard Exploit Protection configured with the following minimum mitigations enabled; Control Flow Guard (CFG), Randomize memory allocations (Bottom-Up ASLR), Export address filtering (EAF), Import address filtering (IAF), and Data Execution Prevention (DEP). The following link describes how to enable Exploit Protection, https://learn.microsoft.com/en-us/microsoft-365/security/defender-endpoint/enable-exploit-protection?view=o365-worldwide. 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). **TD0823 has been applied.**
Test Steps	• Verify the Tamper protection configuration on the agent_config.json file. • Stop the agent process in cmd using command "sc stop xagt" and verify with "sc query xagt". • Go to Windows Security, select "App and browser control" then Exploit protection > Program Settings and select Add program to customize > Choose exact file path and look for xagt.exe in C:\ProgramFiles (x86)\FireEye\xagt. • Enable mitigations Control Flow Guard (CFG), Randomize memory allocations (Bottom-Up ASLR), Export address filtering (EAF), Import address filtering (IAF), and Data Execution Prevention (DEP).

	• Verify that the agent is able to run after mitigations have been enabled by running command “sc start xagt” in command prompt.
Expected Test Results	The TOE should run successfully after minimum mitigations have been enabled on Windows Defender Exploit Guard.
Pass/Fail with Explanation	PASS. The TOE runs successfully after minimum mitigations have been enabled on Windows Defender Exploit Guard. This meets the testing requirements.

6.1.47 FPT_AEX_EXT.1.4 Test #1

Item	Data
Test Assurance Activity	The evaluator shall run the application and determine where it writes its files. For files where the user does not choose the destination, the evaluator shall check whether the destination directory contains executable files. This varies per platform: **Platforms: Microsoft Windows...** For Windows Universal Applications the evaluator shall consider the requirement met because the platform forces applications to write all data within the application working directory (sandbox). For Windows Desktop Applications the evaluator shall run the program, mimicking normal usage, and note where all user-modifiable files are written. The evaluator shall ensure that there are no executable files stored in the same directories to which the application wrote user-modifiable files.
Test Steps	• Identify the PID of xagt.exe in Windows Task Manager. • Start ProcMon tool with filter for the PID and the WriteFile operation. • Log into the HX Server and navigate to Admin>Policies>Agent Default Policy>Malware Scans. Under Scan Settings toggle User Canceled Scans and save, toggle it back and save again. • On the TOE platform, observe the ProcMon tool and note the path where files are written. • Browse to the path of the files and verify that there are no executables.
Expected Test Results	No executable files will be stored in the path where the TOE writes user-modifiable files and no data files are stored in the application’s install directory.
Pass/Fail with Explanation	PASS. No executable files are stored in the path where the TOE writes user-modifiable files and no data files are stored in the application’s install directory. This meets the testing requirements.

6.1.48 FPT_AEX_EXT.1.5 Test #1

Item	Data

Test Assurance Activity	The evaluator will inspect every native executable included in the TOE to ensure that stack-based buffer overflow protection is present.
	Platforms:Microsoft Windows...
	Applications that run as Managed Code in the .NET Framework do not require these stack protections. Applications developed in Object Pascal using the Delphi IDE compiled with RangeChecking enabled comply with this element. For other code, the evaluator shall review the TSS and verify that the /GS flag was used during compilation. The evaluator shall run a tool like, BinSkim, 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. **TD0815 has been applied**
Test Steps	• Review the TSS and verify that the /GS flag was used. • Verify that the TOE is running using command prompt with “sc query xagt”. • Use the python pefile module to read the TOE executable and verify the use of the /GS flag.
Expected Test Results	A security cookie will be present in the TOE configuration which ensures the use of /GS flag.
Pass/Fail with Explanation	PASS. The use of /GS flag on the TOE was verified using the python pefile. Therefore, the stack-based buffer overflow protection is present in the TOE. This meets the testing requirements.

6.1.49 FPT_API_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall then compare the list with the supported APIs (available through e.g. developer accounts, platform developer groups) and ensure that all APIs listed in the TSS are supported.
Test Steps	• Verify that the TSS lists the platform APIs used in the application. • 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.
Expected Test Results	• TSS listed with the platform APIs used in the application. • Screenshot evidence of the supported APIs (available through e.g. developer accounts, platform developer groups).
Pass/Fail with Explanation	PASS. All APIs listed in the TSS are documented and supported by Microsoft. This meets the testing requirements.

6.1.50 FPT_IDV_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall install the application, then check for the existence of version information. If SWID tags is selected the evaluator shall check for a .swidtag file. The evaluator shall open the file and verify that is contains at least a SoftwareIdentity element and an Entity element.
Test Steps	• Locate the swidtag file of the TOE. • Verify that the swidtag file contains a SoftwareIdentity element and an Entity element.
Expected Test Results	The swidtag file should contain a SoftwareIdentity element and an Entity element.
Pass/Fail with Explanation	PASS. The swidtag file contains a SoftwareIdentity element and an Entity element. This meets the testing requirement.

6.1.51 FPT_LIB_EXT.1.1 Test #1

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

Test Steps	• Survey the installation directory for dynamic libraries. • Cross-check and verify that the .dll files in the directory match those in the ST.
Expected Test Results	The dynamic libraries installed in the TOE match those listed in the ST.
Pass/Fail with Explanation	PASS. The dynamic libraries installed in the TOE match those listed in the ST. This meets the testing requirements.

6.1.52 FPT_TUD_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall check for an update using procedures described in either the application documentation or the platform documentation and verify that the application does not issue an error. If it is updated or if it reports that no update is available this requirement is considered to be met.
Test Steps	• Login to the HX Server web console. • Navigate to Admin > Agent Upgrade and show the list of Agent versions. • Download the latest update from the agent version. • Install the TOE with the setup file. • Verify the version of the TOE.
Expected Test Results	TOE should get installed with the latest version available.
Pass/Fail with Explanation	PASS. The TOE is installed with the latest version available. This meets the testing requirements.

6.1.53 FPT_TUD_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall query the application for the current version of the software according to the operational user guidance. The evaluator shall then verify that the current version matches that of the documented and installed version.
Test Steps	• Verify the TOE version on the command prompt. • Verify the version details of TOE in the command prompt matches with that mentioned in ST.
Expected Test Results	The installed version of the TOE matches the one listed in the ST
Pass/Fail with Explanation	PASS. The application's current version matches that of the documented and installed version. This meets the testing requirements.

6.1.54 FPT_TUD_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	The evaluator shall verify that the application's executable files are not changed by the application. **For all other platforms**, the evaluator shall perform the following test: 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.
Test Steps	• On the host machine, locate all TOE executable files in the TOE install directory (C:\Program Files(x86)\FireEye\xagt) using HashMyFiles tool.

	• With the HashMyFiles tool, save the hash of all discovered executables files as a text file. • Login to the HX Server web console and navigate to “Hosts>All Hosts and select TOE Host Machine, Select Acquire and run all listed acquisitions. • On the HX Server, navigate to Acquisitions to verify that they have been run and completed. • Re-generate hash files for the TOE executables as a text file using HashMyFiles. • Compare both text files using WinMerge and verify that the hash values have not changed.
Expected Test Results	The hash values of the TOE executable prior using the TOE do not change after using the TOE.
Pass/Fail with Explanation	PASS. The hash values of the TOE executable do not change after using features of the TOE. This meets the testing requirements.

6.1.55 FPT_TUD_EXT.1.5 TSS #1

Item	Data
Test Assurance Activity	The evaluator shall verify that the TSS identifies how the application is distributed. If "**with the platform**" is selected the evaluated shall perform a clean installation or factory reset to confirm that TOE software is included as part of the platform OS. If "**as an additional package**" is selected the evaluator shall perform the tests in FPT_TUD_EXT.2.
Pass/Fail with Explanation	PASS, the ST selects "**as an additional package**" and hence the FPT_TUD_EXT.2 is performed.

6.1.56 FPT_TUD_EXT.2.1 Test #1

Item	Data
Test Assurance Activity	If a container image is claimed the evaluator shall verify that application updates are distributed as container images. If the format of the platform-supported package manager is claimed, *t*he evaluator shall verify that application updates are distributed in the format supported by the platform. This varies per platform: **Platforms:Microsoft Windows...** The evaluator shall ensure that the application is packaged in the standard Windows Installer (.MSI) format, the Windows Application Software (.EXE) format signed using the Microsoft Authenticode process, or the Windows Universal Application package (.APPX) format. See https://msdn.microsoft.com/en-us/library/ms537364(v=vs.85).aspx for details regarding Authenticode signing. **TD0628 has been applied.**
Test Steps	• Locate the TOE installation package, verify its properties. • Under the “General” tab, verify that it is a .msi format.
Expected Test Results	The TOE install file is a .msi format.
Pass/Fail with Explanation	PASS. The TOE installation file is a standard Windows Installer Package (.msi). This meets the testing requirements.

6.1.57 FPT_TUD_EXT.2.2 Test #1

Item	Data
Test Assurance Activity	**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. **TD0664 has been applied**
Test Steps	• Before installing the TOE, in command prompt under root directory C:\ run command "dir /B /S > before_install.txt" to record the path of every file on the entire filesystem and redirect it to a text file. • Install the TOE and verify that it is running using "sc query xagt" in cmd. • Uninstall the TOE. • After uninstalling the TOE, in command prompt under root directory C:\ run command "dir /B /S > after_uninstall.txt" to record the path of every file on the entire filesystem and redirect it to a text file. • Compare the text files using WinMerge software and verify that no files, other than configuration, output and audit/log files have been added to filesystem.
Expected Test Results	No files, other than configuration, output, and audit/log files, will be added to the filesystem.
Pass/Fail with Explanation	PASS. No files, other than configuration, output, and audit/log files, have been added to the filesystem. This meets the testing requirements.

6.1.58 FTP_DIT_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	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 Steps	• Initiate an HTTPS connection to the HX Server from the TOE by running the application. • Observe the network traffic on the packet capture and verify that it is encrypted by TLS or HTTPS.
Expected Test Results	Traffic between the TOE and the HX Server is encrypted with TLS.
Pass/Fail with Explanation	PASS. Traffic between the TOE and HX server is encrypted with TLS. This meets the testing requirements.

6.1.59 FTP_DIT_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	PASS. This test is performed in conjunction with FTP_DIT_EXT.1.1 Test #1 where the traffic between the TOE and HX Server is verified to be encrypted with TLS.

6.1.60 FTP_DIT_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	N/A. The TOE does not transmit user credentials.

6.2 TLSC-MA

6.2.1 FCS_TLSC_EXT.1 Test #1

Item	Data
Test Assurance Activity	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 Steps	• Attempt a TLS connection from TOE to the TLS server and show the connection being successful. • Verify with packet capture that the TOE can successfully establish a connection with the cipher suite specified by the requirement: TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246.
Expected Test Results	Packet capture evidence showing successful TLS connection with the claimed cipher suite.
Pass/Fail with Explanation	PASS. The TOE successfully negotiates the claimed cipher suite. This meets the testing requirements.

6.2.2 FCS_TLSC_EXT.1 Test #2

Item	Data
Test Assurance Activity	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 Steps	<u>Valid Certificate</u> • Create a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension.

	• Attempt a TLS connection from the TOE to the TLS Server and show the connection being successful. • Verify the successful packet capture showing the Server Authentication purpose enabled. Invalid Certificate • Create a server certificate that lacks the Server Authentication purpose in the extendedKeyUsage extension. • Attempt a TLS connection from the TOE to the TLS Server and show the connection being unsuccessful. • Verify the unsuccessful packet capture lacking the Server Authentication purpose.
Expected Test Results	• Packet capture evidence showing successful TLS connection with the Server Authentication purpose enabled. • Packet capture evidence showing unsuccessful TLS connection with lacking the Server Authentication purpose.
Pass/Fail with Explanation	PASS. The TOE accepts the connection with serverAuth Extended Key Usage in the server certificate and denies the connection to a server with a lack of serverAuth Extended Key Usage in the server certificate. This meets the testing requirements.

6.2.3 FCS_TLSC_EXT.1 Test #3

Item	Data
Test Assurance Activity	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 Steps	• Create an ECDSA server certificate. • Attempt a TLS connection from the TOE to the TLS Server and show the connection being unsuccessful. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful TLS connection with ECDSA server certificate.
Pass/Fail with Explanation	PASS. The TOE denies a connection when a server certificate does not match the server-selected cipher suite. This meets the testing requirements.

6.2.4 FCS_TLSC_EXT.1 Test #4

Item	Data
Test Assurance Activity	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 Steps	• Attempt a TLS connection using "Acumen-TLSC" tool with the server using the TLS_NULL_WITH_NULL_NULL cipher suite and verify the connection fails. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful TLS connection with TLS_NULL_WITH_NULL_NULL cipher suite.
Pass/Fail with Explanation	PASS. The TOE denies a connection when a server selects the TLS_NULL_WITH_NULL_NULL cipher suite. This meets the testing requirements.

6.2.5 FCS_TLSC_EXT.1 Test #5.1

Item	Data

Test Assurance Activity	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 Steps	• Attempt a TLS connection using "Acumen-TLSC" tool with an undefined TLS version and verify the connection fails. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful TLS connection with an undefined TLS version.
Pass/Fail with Explanation	PASS. The TOE denies a connection when a server selects an undefined TLS version. This meets the testing requirements.

6.2.6 FCS_TLSC_EXT.1 Test #5.2

Item	Data
Test Assurance Activity	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 Steps	• Attempt a TLS connection using "Acumen-TLSC" tool with the most recent unsupported TLS version and verify the connection fails. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful TLS connection with the most recent unsupported TLS version.
Pass/Fail with Explanation	PASS. The TOE denies a connection when a server selects the most recent unsupported TLS version 1.1. This meets the testing requirements.

6.2.7 FCS_TLSC_EXT.1 Test #5.3

Item	Data
Test Assurance Activity	[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.
Pass/Fail with Explanation	N/A. DHE or ECDHE cipher suites are not supported as per the ST.

6.2.8 FCS_TLSC_EXT.1 Test #5.4

Item	Data
Test Assurance Activity	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 Steps	• Attempt a TLS connection using "Acumen-TLSC" tool to modify the server's selected cipher suite in the Server handshake message and verify that the connection fails. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful TLS connection when a server's selected cipher suite not presented in the Client Hello handshake message.
Pass/Fail with Explanation	PASS. The TOE denies a connection when a server's selected cipher suite is not presented in the Client Hello handshake message. This meets the testing requirements.

6.2.9 FCS_TLSC_EXT.1 Test #5.5

Item	Data
Test Assurance Activity	[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.
Pass/Fail with Explanation	N/A. DHE or ECDHE cipher suites are not supported as per the ST.

6.2.10 FCS_TLSC_EXT.1 Test #5.6

Item	Data
Test Assurance Activity	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 Steps	• Attempt a TLS connection using "Acumen-TLSC" tool to modify a byte in the Server Finished handshake message and verify that the connection fails. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful TLS connection when a byte is modified in the Server Finished handshake message.
Pass/Fail with Explanation	PASS. The TOE denies a connection when a byte is modified in the Server Finished handshake message. This meets the testing requirements.

6.2.11 FCS_TLSC_EXT.1 Test #5.7

Item	Data
Test Assurance Activity	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.
Test Steps	• Attempt a TLS connection using "Acumen-TLSC" tool to send a garbled message after the Change Cipher Spec message is issued and verify that the connection fails. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful TLS connection when a garbled message is sent after the Change Cipher Spec message is issued.
Pass/Fail with Explanation	PASS. The TOE denies a connection when a message consisting of random bytes is sent from the server after the server has issued the Change Cipher Spec message. This meets the testing requirements.

6.2.12 FCS_TLSC_EXT.1 Test #6

Item	Data
Test Assurance Activity	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 6: 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. **TD0499 has been applied.**

Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Note: The TOE only supports Fully Qualified Domain Names (FQDN) as reference identifiers in either the SAN extension or the CN field of the presented TLS server X.509 certificate. • Configure the reference identifier on the TOE. • Create a Server Certificate with no SAN extension and a FQDN in CN field that does not match the reference identifier. • Attempt a TLS connection and verify that the connection fails. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful TLS connection when a server certificate with no SAN extension and a FQDN in CN field that does not match the reference identifier is presented.
Pass/Fail with Explanation	PASS. The TOE denies a connection when a server certificate with no SAN extension and a FQDN in CN field that does not match the reference identifier is presented. This meets the testing requirements.

6.2.13 FCS_TLSC_EXT.1 Test #7

Item	Data
Test Assurance Activity	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 7: 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. **TD0499 has been applied.**
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Note: The TOE only supports Fully Qualified Domain Names (FQDN) as reference identifiers in either the SAN extension or the CN field of the presented TLS server X.509 certificate. • Configure the reference identifier on the TOE. • Create a Server Certificate with a FQDN in CN field that matches the reference identifier and a FQDN in SAN extension that does not match the reference identifier. • Attempt a TLS connection and verify that the connection fails. • Verify the unsuccessful connection with the packet capture.

Expected Test Results	Packet capture evidence showing unsuccessful TLS connection when a server certificate with a FQDN in CN field that matches the reference identifier and a FQDN in SAN extension that does not match the reference identifier is presented.
Pass/Fail with Explanation	PASS. The TOE denies a connection when a server certificate with a FQDN in CN field that matches the reference identifier and a FQDN in SAN extension that does not match the reference identifier is presented. This meets the testing requirements.

6.2.14 FCS_TLSC_EXT.1 Test #8

Item	Data
Test Assurance Activity	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 8: [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. **TD0499 has been applied.**
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Note: The TOE only supports Fully Qualified Domain Names (FQDN) as reference identifiers in either the SAN extension or the CN field of the presented TLS server X.509 certificate. • Configure the reference identifier on the TOE. • Create a Server Certificate with a FQDN in CN field that matches the reference identifier and no SAN extension. • Attempt a TLS connection and verify that the connection succeeds. • Verify the successful connection with the packet capture.
Expected Test Results	Packet capture evidence showing successful TLS connection when a server certificate with no SAN extension and a FQDN in CN field that matches the reference identifier is presented.
Pass/Fail with Explanation	PASS. The TOE accepts the connection when a server certificate with no SAN extension and a FQDN in CN field that matches the reference identifier is presented. This meets the testing requirements.

6.2.15 FCS_TLSC_EXT.1 Test #9

Item	Data
Test Assurance Activity	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 9: 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. **TD0499 has been applied.**
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Note: The TOE only supports Fully Qualified Domain Names (FQDN) as reference identifiers in either the SAN extension or the CN field of the presented TLS server X.509 certificate. • Configure the reference identifier on the TOE. • Create a Server Certificate with a FQDN in CN field that does not match the reference identifier and a FQDN in SAN extension that matches the reference identifier. • Attempt a TLS connection and verify that the connection succeeds. • Verify the successful connection with the packet capture.
Expected Test Results	• Packet capture evidence showing successful TLS connection when a server certificate with a FQDN in CN field that does not match the reference identifier and a FQDN in SAN extension that matches the reference identifier is presented.
Pass/Fail with Explanation	PASS. The TOE accepts the connection when a server certificate with a FQDN in CN field that does not match the reference identifier and a FQDN in SAN extension that matches the reference identifier is presented. This meets the testing requirements.

6.2.16 FCS_TLSC_EXT.1 Test #10.1

Item	Data
Test Assurance Activity	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. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. **Test 10.1:** [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. **TD0499 has been applied.**
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Note: The TOE only supports Fully Qualified Domain Names (FQDN) as reference identifiers in either the SAN extension or the CN field of the presented TLS server X.509 certificate. CN

	• Configure the reference identifier on the TOE. • Create a Server Certificate containing a wildcard that is not in the left-most label of the presented identifier in CN. • Attempt a TLS connection and verify that the connection fails. • Verify the unsuccessful connection with the packet capture. <u>SAN</u> • Create a Server Certificate containing a wildcard that is not in the left-most label of the presented identifier in SAN. • Attempt a TLS connection and verify that the connection fails. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful connection when a server certificate containing a wildcard in the CN field or SAN extension, which is not in the left-most label is presented.
Pass/Fail with Explanation	PASS. The TOE rejects the connection when a server certificate containing a wildcard in the CN field or SAN extension is presented; however, the wildcard was not in the left-most label. This meets the testing requirements.

6.2.17 FCS_TLSC_EXT.1 Test #10.2(a)

Item	Data
Test Assurance Activity	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. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 10.2: [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. **TD0499 has been applied.**
Test Steps	<u>Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6.</u> <u>Note: The TOE only supports Fully Qualified Domain Names (FQDN) as reference identifiers in either the SAN extension or the CN field of the presented TLS server X.509 certificate.</u> <u>FQDN in CN field</u> • Configure the reference identifier with a single left-most label on the TOE. • Create a Server Certificate containing a wildcard in the left-most label but not preceding the public suffix in CN. • Attempt a TLS connection and verify that the connection succeeds. • Verify the successful connection with the packet capture.

	FQDN in SAN extension • Create a Server Certificate containing a wildcard in the left-most label but not preceding the public suffix in SAN. • Attempt a TLS connection and verify that the connection succeeds. • Verify the successful connection with the packet capture.
Expected Test Results	Packet capture evidence showing successful connection when a server certificate containing a wildcard in the CN field or SAN extension, positioned in the left-most label but not preceding the public suffix, is presented, however the TOE is configured with a reference identifier having a single left-most label.
Pass/Fail with Explanation	PASS. The TOE accepts the connection to a TLS server when the evaluator configures the TOE with a reference identifier with a single left-most label, and the presented TLS server certificate contains a wildcard in the CN field or SAN extension. Specifically, the wildcard is positioned in the left-most label but does not precede the public suffix. This meets the testing requirements.

6.2.18 FCS_TLSC_EXT.1 Test #10.2(b)

Item	Data
Test Assurance Activity	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. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 10.2: [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 without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Note: The TOE only supports Fully Qualified Domain Names (FQDN) as reference identifiers in either the SAN extension or the CN field of the presented TLS server X.509 certificate. FQDN in CN field • Configure the reference identifier on the TOE without a left-most label. • Create a Server Certificate containing a wildcard in the left-most label but not preceding the public suffix in CN field. • Attempt a TLS connection and verify that the connection fails. • Verify the unsuccessful connection with the packet capture. FQDN in SAN extension

	• Create a Server Certificate containing a wildcard in the left-most label but not preceding the public suffix in SAN Extension. • Attempt a TLS connection and verify that the connection fails. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful connection when a server certificate containing a wildcard, in the CN field or SAN extension, positioned in the left-most label but not preceding the public suffix, is presented, however the TOE's reference identifier is constructed without a left-most label as in the certificate.
Pass/Fail with Explanation	PASS. The TOE rejects the connection when a server certificate containing a wildcard in the CN field or SAN extension, positioned in the left-most label but not preceding the public suffix, is presented. However, the TOE's reference identifier is constructed without a left-most label as in the certificate. This meets the testing requirements.

6.2.19 FCS_TLSC_EXT.1 Test #10.2(c)

Item	Data
Test Assurance Activity	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. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 10.2: [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 two left-most labels (e.g. bar.foo.example.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Note: The TOE only supports Fully Qualified Domain Names (FQDN) as reference identifiers in either the SAN extension or the CN field of the presented TLS server X.509 certificate. Wildcard in CN field • Configure the reference identifier with two left-most labels on the TOE. • Create a Server Certificate containing a wildcard in the left-most label but not preceding the public suffix in CN. • Attempt a TLS connection and verify that the connection fails. • Verify the unsuccessful connection with the packet capture. Wildcard in SAN extension • Create a Server Certificate containing a wildcard in the left-most label but not preceding the public suffix in CN. • Attempt a TLS connection and verify that the connection fails.

	• Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful connection when a server certificate containing a wildcard, in the CN field or SAN extension, positioned in the left-most label but not preceding the public suffix, is presented. however, the TOE's reference identifier is constructed with two left-most label.
Pass/Fail with Explanation	PASS. The TOE rejects the connection when a server certificate containing a wildcard in the CN field or SAN extension, positioned in the left-most label but not preceding the public suffix, is presented. However, the TOE's reference identifier is constructed with left-most label. This meets the testing requirements.

6.2.20 FCS_TLSC_EXT.1 Test #10.3(a)

Item	Data
Test Assurance Activity	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. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 10.3: [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. **TD0499 has been applied.**
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Note: The TOE only supports Fully Qualified Domain Names (FQDN) as reference identifiers in either the SAN extension or the CN field of the presented TLS server X.509 certificate. Wildcard in CN field • Configure the reference identifier on the TOE with a single left-most label. • Create a Server Certificate containing a wildcard in the left-most label immediately preceding the public suffix in CN. • Attempt a TLS connection and verify that the connection fails. • Verify the unsuccessful connection with the packet capture. Wildcard in SAN extension • Create a Server Certificate containing a wildcard in the left-most label immediately preceding the public suffix in SAN. • Attempt a TLS connection and verify that the connection fails. • Verify the unsuccessful connection with the packet capture.

Expected Test Results	Packet capture evidence showing an unsuccessful connection when a server certificate containing a wildcard in the CN field or SAN extension, positioned in the left-most label immediately preceding the public suffix, is presented. However, the TOE is configured with a reference identifier having a single left-most label.
Pass/Fail with Explanation	PASS. The TOE rejects the connection when a server certificate containing a wildcard in the CN field or SAN extension, positioned in the left-most label immediately preceding the public suffix, is presented. However, the TOE is configured with a reference identifier having a single left-most label. This meets the testing requirements.

6.2.21 FCS_TLSC_EXT.1 Test #10.3(b)

Item	Data
Test Assurance Activity	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. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 10.3: [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 two left-most labels (e.g. bar.foo.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Note: The TOE only supports Fully Qualified Domain Names (FQDN) as reference identifiers in either the SAN extension or the CN field of the presented TLS server X.509 certificate. Wildcard in CN field • Configure the reference identifier on the TOE with two left-most labels. • Create a Server Certificate containing a wildcard in the left-most label immediately preceding the public suffix in CN. • Attempt a TLS connection and verify that the connection fails. • Verify the unsuccessful connection with the packet capture. Wildcard in SAN extension • Create a Server Certificate containing a wildcard in the left-most label immediately preceding the public suffix in SAN. • Attempt a TLS connection and verify that the connection fails. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing Unsuccessful TLS connection when a server certificate containing a wildcard, in the CN field or SAN extension, positioned in the left-most label

	immediately preceding the public suffix, is presented, however, the TOE is configured with a reference identifier with two left-most labels.
Pass/Fail with Explanation	PASS. The TOE rejects the connection when a server certificate containing a wildcard, in the CN field or SAN extension, positioned in the left-most label immediately preceding the public suffix, is presented. However, the TOE is configured with a reference identifier having two left-most labels. This meets the testing requirements.

6.2.22 FCS_TLSC_EXT.1 Test #10.4

Item	Data
Test Assurance Activity	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. The evaluator shall perform the following wildcard tests with each supported type of reference identifier. Test 10.4: [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. **TD0499 has been applied.**
Pass/Fail with Explanation	N/A. As the TOE does not support certificate pinning; and the TOE supports wildcards.

6.2.23 FCS_TLSC_EXT.1 Test #11

Item	Data
Test Assurance Activity	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 11: [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. **TD0499 has been applied.**
Pass/Fail with Explanation	N/A. As the TOE does not support certificate pinning; and the TOE does not support URI or Service name reference identifiers.

6.2.24 FCS_TLSC_EXT.1 Test #12

Item	Data
Test Assurance Activity	Test 12: [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.

Pass/Fail with Explanation	N/A. The TOE does not support pinned certificates.

6.2.25 FCS_TLSC_EXT.1 Test #13a

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 13: The evaluator shall demonstrate that a server using a certificate with a valid certification path successfully connects. **TD0513 has been applied.**
Test Steps	• Create a valid certificate chain. • Ensure that the TOE contains the valid CA certificate in its config file. • Attempt a TLS connection from the TOE to the TLS Server using a certificate with a valid certification path and show the connection being successful. • Verify the successful connection with the packet capture.
Expected Test Results	Packet capture evidence showing successful TLS connection when a certificate with valid certification path is presented.
Pass/Fail with Explanation	PASS. The TOE accepts the connection when a certificate with valid certification path is presented. This meets the testing requirements.

6.2.26 FCS_TLSC_EXT.1 Test #13b

Item	Data
Test Assurance Activity	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. **TD0513 has been applied.**
Test Steps	• Create an invalid certificate chain. • Attempt a TLS connection from the TOE to the TLS Server using a certificate with an invalid certification path and show the connection being unsuccessful. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful TLS connection when a certificate with an invalid certification path is presented.
Pass/Fail with Explanation	PASS. The TOE denies the connection when a certificate with an invalid certification path is presented. This meets the testing requirements.

6.2.27 FCS_TLSC_EXT.1 Test #13c

Item	Data
Test Assurance Activity	[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. **TD0513 has been applied.**
Pass/Fail with Explanation	N/A. The TOE trust store cannot be managed.

6.2.28 FCS_TLSC_EXT.1 Test #14

Item	Data

Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 14: The evaluator shall demonstrate that a server using a certificate which has been revoked results in an authentication failure.
Pass/Fail with Explanation	Pass. Refer to FIA_X509_EXT.1.1 Test #3 for testing with revoked server certificate.

6.2.29 FCS_TLSC_EXT.1 Test #15

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 15: The evaluator shall demonstrate that a server using a certificate which has passed its expiration date results in an authentication failure.
Test Steps	• Create an expired server certificate. • Attempt a TLS connection from the TOE to the TLS Server using the expired server certificate and show the connection being unsuccessful. • Verify the unsuccessful connection using a collected packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful TLS connection when an expired server certificate is presented.
Pass/Fail with Explanation	PASS. The TOE denies the connection when an expired certificate is presented. This meets the testing requirements.

6.2.30 FCS_TLSC_EXT.1 Test #16

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate (unless excepted) results in the function failing as follows, unless excepted: Test 16: The evaluator shall demonstrate that a server using a certificate which does not have a valid identifier results in an authentication failure.
Test Steps	• Configure the reference identifier on the TOE. • Create a Server Certificate with an invalid reference identifier. • Attempt a TLS connection and verify that the connection fails. • Verify the unsuccessful connection with the packet capture.
Expected Test Results	Packet capture evidence showing unsuccessful TLS connection when a certificate with invalid identifier is presented.
Pass/Fail with Explanation	PASS. The TOE denies the connection when a certificate with invalid identifier is presented. This meets the testing requirements.

6.2.31 FCS_TLSC_EXT.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall establish a connection to a server that is not configured for mutual authentication (i.e. does not send Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE did not send Client's Certificate message (type 11) during handshake.
Test Steps	• Attempt a TLS connection using "Acumen-TLSC" tool for non-mutual authentication and show the connection being successful. • Verify the successful connection with the packet capture and ensure that the server does not send client "Certificate Request (type 13) message" and the TOE does not send Client's Certificate message (type 11).

Expected Test Results	• Packet capture evidence showing successful TLS connection. • Packet capture evidence showing that no client's certificate packets are sent during the handshake when the TLS server is not configured for mutual authentication.
Pass/Fail with Explanation	PASS. The TOE does not send Client's Certificate message (type 11) during the non-mutual authentication. This meets the testing requirements.

6.2.32 FCS_TLSC_EXT.2 Test #2

Item	Data
Test Assurance Activity	The evaluator shall establish a connection to a server with a shared trusted root that is configured for mutual authentication (i.e. it sends Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message.
Test Steps	• Attempt a TLS connection using "Acumen-TLSC" tool for mutual authentication and show the connection being successful. • Verify the successful connection with the packet capture and ensure that the server sends Server's Certificate Request (type 13) message. • Verify that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message.
Expected Test Results	• Packet capture evidence showing successful TLS connection. • Packet capture evidence showing that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message for the Server's Certificate Request (type 13) message.
Pass/Fail with Explanation	PASS. The TOE responds to the Server's Certificate Request (type 13) message with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message for mutual authentication. This meets the testing requirements.

6.2.33 FCS_TLSC_EXT.3.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure the server to send a certificate in the TLS connection that is not supported according to the Client's HashAlgorithm enumeration within the signature_algorithms extension (for example, send a certificate with a SHA-1 signature). The evaluator shall verify that the product disconnects after receiving the server's Certificate handshake message.
Pass/Fail with Explanation	N/A, as this SFR is not claimed in ST.

6.2.34 FCS_TLSC_EXT.3.1 Test #2

Item	Data
Test Assurance Activity	[conditional] If **the client supports a DHE or ECDHE cipher suite**, the evaluator shall configure the server to send a Key Exchange handshake message including a signature not supported according to the client's HashAlgorithm enumeration (for example, the server signed the Key Exchange parameters using a SHA-1 signature). The evaluator shall verify that the product disconnects after receiving the server's Key Exchange handshake message.
Pass/Fail with Explanation	N/A, as this SFR is not claimed in ST.

6.2.35 FCS_TLSC_EXT.4.1 Test #1

Item	Data

Test Assurance Activity	The evaluator shall use a network packet analyzer/sniffer to capture the traffic between the two TLS endpoints. The evaluator shall verify that either the "renegotiation_info" field or the SCSV cipher suite is included in the ClientHello message during the initial handshake.
Test Steps	• Attempt a TLS connection from the TOE to the TLS Server. • Verify the SCSV Cipher Suite in the packet capture.
Expected Test Results	Packet capture evidence showing SCSV cipher suite included in the TOE's ClientHello message.
Pass/Fail with Explanation	PASS. The SCSV cipher suite is included in the TOE's ClientHello message during the initial handshake. This meets the testing requirements.

6.2.36 FCS_TLSC_EXT.4.1 Test #2

Item	Data
Test Assurance Activity	The evaluator shall verify the Client's handling of ServerHello messages received during the initial handshake that include the "renegotiation_info" extension. The evaluator shall modify the length portion of this field in the ServerHello message to be non-zero and verify that the client sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.
Test Steps	• Attempt a TLS connection using "Acumen-TLSC" tool to modify the length portion of the "renegotiation_info" field in the ServerHello message to be non-zero and verify that the connection fails. • Verify the unsuccessful connection with the packet capture. • Attempt a TLS connection using "Acumen-TLSC" tool with a properly formatted field without any modification. • Verify the successful connection with the packet capture.
Expected Test Results	• Packet capture evidence showing unsuccessful connection when the length portion of the "renegotiation_info" field in the ServerHello message is modified to be non-zero. • Packet capture evidence showing successful connection with a properly formatted field without any modification.
Pass/Fail with Explanation	PASS. The TOE terminates the connection when the length portion of the "renegotiation_info" field in the ServerHello message is modified to be non-zero and accepts the connection with a properly formatted field.

6.2.37 FCS_TLSC_EXT.4.1 Test #3

Item	Data
Test Assurance Activity	The evaluator shall verify that ServerHello messages received during secure renegotiation contain the "renegotiation_info" extension. The evaluator shall modify either the "client_verify_data" or "server_verify_data" value and verify that the client terminates the connection.
Test Steps	• Attempt a TLS secure renegotiation connection. • Verify that the ServerHello messages received during secure renegotiation contain the "renegotiation_info" extension. • Attempt to modify the "sever_verify_data" value and verify that the connection fails. • Verify with packet capture that the ServerHello contains the "renegotiation_info" extension. • Using a decrypted packet capture. Verify the TOE's unsuccessful connection after modification of the TLS server's "server_verify_data" field.

Expected Test Results	The TOE will terminate a TLS connection when the "server_verify_data" value in the "renegotiation_info" extension in a ServerHello message is modified.
Pass/Fail with Explanation	PASS. The TOE terminates a TLS connection when the "server_verify_data" value in the "renegotiation_info" extension in a ServerHello message is modified. This meets the testing requirements.

6.2.38 FCS_TLSC_EXT.5.1 Test #1

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	N/A, as this SFR is not claimed in ST.

7 Security Assurance Requirements

7.1 ADV: Developpment

7.1.1 ADV_FSP.1 Basic Functional Specification

7.1.1.1 ADV_FSP.1 TSS 1

Objective	There are no specific assurance activities associated with these SARs, except ensuring the information is provided. The functional specification documentation is provided to support the evaluation activities described in Section 5.1, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because there is insufficient interface information, then an adequate functional specification has not been provided.
Evaluator Findings	The evaluator found that the outlined Assurance Activities have yielded ample information for establishing the content of the TSS section and executing the assurance tasks. Given their direct correlation with the SFRs and their implicit completion, no further documentation or analysis is required.
Verdict	Pass

7.2 AGD: Guidance Documentation

7.2.1 AGD_OPE.1 Operational User Guidance

7.2.1.1 AGD_OPE.1 Guidance 1

Objective	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.
Evaluator Findings	The evaluator examined the section titled **Common Criteria Settings** in the [AGD] and verified that it contains instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE and provides a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.2.1.2 AGD_OPE.1 Guidance 2

Objective	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 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.
Evaluator Findings	The evaluator examined the section titled **Verifying Updates** in the [AGD] describes the process for verifying updates to the TOE by verifying a digital signature. This section also includes instructions for obtaining the update and initiating the update process. The section titled **Overview** describes the scope of functionality under evaluation. It states "**This TOE is an enterprise-managed agent that runs in the background of an endpoint platform. End users will have no interaction with the software and will not be alerted of communications with the external Endpoint Security server. As all interactions take place on the Endpoint Security platform. The administrator is expected to deploy the installer with a compliant agent_config.json file.**" Upon investigation, the evaluator found that the [AGD] document contains the section titled **Excluded Functionality** , which lists the not-evaluated functionalities of the TOE. The evaluator found that [DG], section titled **Installation and Deployment**, talks about about how to obtain the TOE first installation image and any other updates. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.2.2 AGD_PRE.1 Preparative Procedures

7.2.2.1 AGD_PRE.1 Guidance 1

Objective	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.
Evaluator Findings	The evaluator examined the section titled **'Evaluation Platforms'** in the [AGD] to verify that it adequately addresses all platforms claimed for the TOE in the ST. Upon investigation, the evaluator found that the [AGD] states that "**Certification evaluation has been performed on the following Windows platforms:** **•Windows 10 Version 21H2 32-bits running on ESXi Hypervisor v7.0 on an Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows 10 Version 1803 32-bits running on ESXi Hypervisor v7.0 on an Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows 10 Version 1903 32-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows 10 Version 1909 LTSC 32-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows 10 Version 2004 32-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows 10 Version 21H2 64-bits running on ESXi Hypervisor v7.0 on an Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows 10 Version 1803 64-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).**

	•Windows 10 Version 1903 64-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell). **•Windows 10 Version 1909 LTSC 64-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows 10 Version 2004 64-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows 11 Version 21H2 64-bits running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows Server 2016 running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows Server 2019 running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows Server 2012 R2 running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows Server 2008 R2 (SP1) running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).** **•Windows Server 2022 running on ESXi Hypervisor v7.0 on Intel Xeon E5-4620 V4 processor (Broadwell).”** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3 ALC: Life-Cycle Support

7.3.1 ALC_CMC.1 Labelling of the TOE

7.3.1.1 ALC_CMC.1 TSS 1

Objective	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.
Evaluator Findings	The evaluator examined the section titled **Security Target and TOE Reference** in the Security Target to verify that the ST contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Upon investigation, the evaluator found that the ST states that the TOE version is **35.31.31.** **In addition, evaluator also checked the AGD guidance and TOE samples received for testing and found that the version number is consistent with that in the ST.** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.1.2 ALC_CMC.1 TSS 2

Objective	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.

Evaluator Findings	The evaluator examined the vendor web site and determined that the TOE is not advertised. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.1.3 ALC_CMC.1 Guidance 1

Objective	Further, the evaluator shall check the AGD guidance to ensure that the version number is consistent with that in the ST.
Evaluator Findings	The evaluator examined the title page of the AGD to verify that the version number is consistent with that in the ST. Upon investigation, the evaluator found that the AGD **is titled "Endpoint Security xAgent Deployment Guide Release 35.31.0" and "Trellix Endpoint Security (HX) Agent v35.31.31 Common Criteria Guidance Supplement" which is consistent with TOE version 35.31.31.** Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.2 ALC_CMS.1 TOE CM Coverage

7.3.2.1 ALC_CMS.1 Guidance 1

Objective	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 evaluation activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component. Life-cycle support is targeted aspects of the developer's life-cycle and instructions to providers of applications for the developer's devices, rather than an in-depth examination of the TSF manufacturer's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it's a reflection on the information to be made available for evaluation. The evaluator shall ensure that the developer has identified (in guidance documentation for application developers concerning the targeted platform) one or more development environments appropriate for use in developing applications for the developer's platform. For each of these development environments, the developer shall provide information on how to configure the environment to ensure that buffer overflow protection mechanisms in the environment(s) are invoked (e.g., compiler flags). The evaluator shall ensure that this documentation also includes an indication of whether such protections are on by default, or have to be specifically enabled.
Evaluator Findings	Upon investigation, the evaluator found that the ST TSS Table 12 for FPT_AEX_EXT.1 states that "The TOE is designed to operate in an environment in which the following security techniques are in effect: – Data execution prevention, – Mandatory address space layout randomization (no memory map to an explicit address), – Structured exception handler overwrite protection,

	– Export address table access filtering, and – Anti-Return Oriented Programming.“ The section also states “During compilation the TOE is built with several flags enabled that check for engineering flaws. The flags used (or not used) are the following: – The TOE is built with the /GS flag enabled. This reduces the possibilities of stack-based buffer overflows in the product." The guidance documentation does not include configuration for buffer overflow protection as it is enabled by default. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.2.2 ALC_CMS.1 Guidance 2

Objective	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.
Evaluator Findings	The evaluator examined the title page of the Trellix Enpoint Security (HX) Agent v35.31.31 Common Criteria Guidance Supplement to verify that it is associated with the TSF using unique identification. Upon investigation, the evaluator found that the guidance documentation states that “**This document serves as an administrative guide for the TOE: Trellix Endpoint Security (HX) agent v35.31.31 evaluated for Common Criteria against the Application Software Protection Profile v1.4 and Functional Package for Transport Layer Security v1.1**.” Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.3 ALC_TSU_EXT.1 Timely Security Updates

7.3.3.1 ALC_TSU_EXT.1 TSS 1

Objective	The evaluator shall verify that the TSS contains a description of the timely security update process used by the developer to create and deploy security updates. The evaluator shall verify that this description addresses the entire application. The evaluator shall also verify that, in addition to the TOE developer’s process, any third-party processes are also addressed in the description. The evaluator shall also verify that each mechanism for deployment of security updates is described.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, ALC_TSU_EXT.1** in the Security Target to verify that the TSS contains a description of the timely security update process that addresses the entire application (including third-party processes) and that each mechanism for deployment of security updates is described. Upon investigation, the evaluator found that the TSS states that “**Users of the TOE should report any security related issues via the Trellix webpage (https://www.trellix.com/en-us/support/fe-support.html [trellix.com]), which provides a secure channel for reporting.** **Software updates/fixes are also provided by the developer via the Trellix webpage. Public availability of an update for a publicly disclosed vulnerability is typically 90 days or less and a maximum of 180 days.”**

	Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.3.2 ALC_TSU.1 TSS 2

Objective	The evaluator shall verify that, for each deployment mechanism described for the update process, the TSS lists a time between public disclosure of a vulnerability and public availability of the security update to the TOE patching this vulnerability, to include any third-party or carrier delays in deployment. The evaluator shall verify that this time is expressed in a number or range of days.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, ALC_TSU_EXT.1** in the Security Target to verify that, for each deployment mechanism described for the update process, the TSS lists a time between public disclosure of a vulnerability and public availability of the security update to the TOE patching this vulnerability. Upon investigation, the evaluator found that the TSS states that "**Public availability of an update for a publicly disclosed vulnerability is typically 90 days or less and a maximum of 180 days.**" Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.3.3.3 ALC_TSU.1 TSS 3

Objective	The evaluator shall verify that this description includes the publicly available mechanisms (including either an email address or website) for reporting security issues related to the TOE. The evaluator shall verify that the description of this mechanism includes a method for protecting the report either using a public key for encrypting email or a trusted channel for a website.
Evaluator Findings	The evaluator examined the section titled **TOE Summary Specification, Table 12, ALC_TSU_EXT.1** in the Security Target to verify that the TSS includes the publicly available mechanisms for reporting security issues related to the TOE, including a method for protecting the report. Upon investigation, the evaluator found that the TSS states that "**Users of the TOE should report any security related issues via the Trellix webpage (https://www.trellix.com/en-us/support/fe-support.html [trellix.com]), which provides a secure channel for reporting**". Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.4 ATE: Tests

7.4.1 ATE_IND.1 Independent Testing – Conformance

Objective	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.
Evaluator Findings	In support of the AAs in the PP, the evaluator created a test plan. This test plan includes an equivalency argument, a description of the test infrastructure (including the host platforms), each test case, and actual results for each test case. Based on these findings, this work unit is considered satisfied
Verdict	Pass

7.5 AVA: Vulnerability Assessment

7.5.1 AVA_VAN.1 Vulnerability Survey

7.5.1.1 AVA_VAN.1 Activity 1 **[Labgram #116]**

Objective	The evaluator shall document their analysis and testing of potential vulnerabilities with respect to this requirement. The evaluator documents the sources consulted and the vulnerabilities found in the report.
Evaluator Findings	The evaluator documented their analysis and testing of potential vulnerabilities with respect to this requirement. Public searches were performed against all keywords found within the Security Target and AGD that may be applicable to specific TOE components. This included protocols, TOE software version, and TOE hardware to ensure sufficient coverage under AVA. The evaluator searched the Internet for potential vulnerabilities in the TOE using the web sites listed below. The sources of publicly available information are provided below. • http://nvd.nist.gov/

	• https://www.cvedetails.com/ The evaluator performed the public domain vulnerability searches using the following key words. The last vulnerability search was performed on May 29, 2024. • Trellix • fireeye • libuv • openssl version 3.0.8 • fips.dll • legacy.dll • libcrypto-3-64.dll • libcrypto-3.dll • libsll-3-64 • libssl-3.dll • zlib 1.2.13 • CryptAcquireContextW, CryptGenRandom, CryptReleaseContext, CryptProtectData, CryptUnprotectData • vcruntime140.dll, vccorlib140.dll, msvcp140.dll, concrt140.dll, ucrtbase.dll, api-ms-win-core-console-l1-1-0.dll, api-ms-win-core-console-l1-2-0.dll, api-ms-win-core-datetime-l1-1-0.dll, api-ms-win-core-debug-l1-1-0.dll, api-ms-win-core-errorhandling-l1-1-0.dll, api-ms-win-core-fibers-l1-1-0.dll, api-ms-win-core-file-l1-1-0.dll, api-ms-win-core-file-l1-2-0.dll, api-ms-win-core-file-l2-1-0.dll, api-ms-win-core-handle-l1-1-0.dll, api-ms-win-core-heap-l1-1-0.dll, api-ms-win-core-interlocked-l1-1-0.dll, api-ms-win-core-libraryloader-l1-1-0.dll, api-ms-win-core-localization-l1-2-0.dll, api-ms-win-core-memory-l1-1-0.dll, api-ms-win-core-namedpipe-l1-1-0.dll, api-ms-win-core-processenvironment-l1-1-0.dll, api-ms-win-core-processthreads-l1-1-0.dll, api-ms-win-core-processthreads-l1-1-1.dll, api-ms-win-core-profile-l1-1-0.dll, api-ms-win-core-rtlsupport-l1-1-0.dll, api-ms-win-core-string-l1-1-0.dll, api-ms-win-core-synch-l1-1-0.dll, api-ms-win-core-synch-l1-2-0.dll, api-ms-win-core-sysinfo-l1-1-0.dll, api-ms-win-core-timezone-l1-1-0.dll, api-ms-win-core-util-l1-1-0.dll, api-ms-win-crt-conio-l1-1-0.dll, api-ms-win-crt-convert-l1-1-0.dll, api-ms-win-crt-environment-l1-1-0.dll, api-ms-win-crt-filesystem-l1-1-0.dll, api-ms-win-crt-heap-l1-1-0.dll, api-ms-win-crt-locale-l1-1-0.dll, api-ms-win-crt-math-l1-1-0.dll, api-ms-win-crt-multibyte-l1-1-0.dll, api-ms-win-crt-private-l1-1-0.dll, api-ms-win-crt-process-l1-1-0.dll, api-ms-win-crt-runtime-l1-1-0.dll, api-ms-win-crt-stdio-l1-1-0.dll, api-ms-win-crt-string-l1-1-0.dll, api-ms-win-crt-time-l1-1-0.dll, api-ms-win-crt-utility-l1-1-0.dll, msvcp140_1.dll, msvcp140_2.dll, msvcp140_atomic_wait.dll, msvcp140_codecvt_ids.dll, vcruntime140_1.dll Based upon the analysis, any issues found were patched in the TOE version and prior versions, mitigating the risk factor. Details can be found in the separate Vulnerability Analysis document. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

7.5.1.2 AVA_VAN.1 Activity 2

Objective	**Conditional 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.

Evaluator Findings	The evaluator documented their analysis and testing of potential malicious files with respect to this requirement. The evaluator performed the virus scans using Windows Defender with the latest virus definitions. The last virus scan was performed on May 17th, 2024. Based upon the analysis, no malicious files were identified. Based on these findings, this assurance activity is considered satisfied.
Verdict	Pass

8 Conclusion

The testing shows that all test cases required for conformance have passed testing.

End of Document