Assurance Activity Report for
Enveil ZeroReveal® Compute Fabric Client v4.6.3

Enveil ZeroReveal® Compute Fabric Client Security Target
Version 2.1

ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1
Version 4.6.3

Protection Profile for Application Software, Version 1.4
Functional Package for Transport Layer Security (TLS), Version 1.1

AAR Version 2.2, April 08, 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:
Enveil Inc.

The Author of the Security Target:
Acumen Security, LLC

The TOE Evaluation was Sponsored by:
Enveil Inc.

Evaluation Personnel:
Acumen Security, LLC
Rupal Gupta
Joan Marshall

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	10/19/2023	Initial Release
0.2	02/05/2024	Changes according to comments
0.3	02/14/2024	Changes as per AGD
1.0	02/20/2024	Added TD0815 and changes as per the AGD
2.0	04/04/2024	Updated based on ECR comments
2.1	04/05/2024	Updated based on 04.05.2024 18:53 Certificate Review Notes.
2.2	04/08/2024	Updated the AVA section based on ECR comment

Contents

1   TOE Overview

The TOE is the Enveil ZeroReveal Compute Fabric Client (otherwise referred to as the ZeroReveal Client or the TOE) software application which communicates to one or more instances of the Enveil ZeroReveal Compute Fabric Client software application via REST API over mutually authenticated HTTPS over TLS.

The TOE is a homomorphic encryption engine for database queries. In normal database operation, a query is submitted in plain text, and a plain text answer retrieved for the querier. While the communication between the querier and the database engine itself may be transmitted through a tunnel such as IPsec, TLS, or SSH, the contents of the query are always in plaintext. The ZeroReveal Compute Fabric Client takes an authenticated user's database query and encrypts it using Enveil's proprietary homomorphic encryption process. This encrypted query is passed via a mutually authenticated TLS trusted channel from ZeroReveal Client to ZeroReveal Server. The encrypted query is never decrypted during this process, which prevents ZeroReveal Server and its owners/administrators from being able to tell what the query was searching for and what items in the database (if any) matched the query. The output of this process is an encrypted response that is sent back to ZeroReveal Client. In this way, the database itself is not strictly aware of what the query was and no individual point in the chain between the user and the information know what was requested.

The ZeroReveal Client (the TOE) and ZeroReveal Server are evaluated separately as software applications only and the homomorphic encryption techniques used for the ZeroReveal Client and ZeroReveal Server operations are outside the scope this evaluation.

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, 07 October 2021 [AppPP] and *Functional Package for Transport Layer Security (TLS)*, Version 1.1, 12 February 2019 [TLSPkg] based upon the core SFRs and those implemented based on selections within the PPs/EPs.

3   Test Equivalency Justification

The TOE implements 4 different cryptographic channels with trusted IT products:

- HTTPS/TLSv1.2 Web Server with or without TLS client authentication – Remote Administration
- HTTPS/TLSv1.2 Web Server with or without TLS client authentication – REST API
- TLSv1.2 client with or without TLS client authentication –Zero Reveal Server
- TLSv1.2 client with or without TLS client authentication – LDAP server

The TOE uses a single TLS implementation in the Bouncy Castle FIPS cryptographic library version 1.0.2. The TOE maintains a global TLS configuration that is enforced for all TLS connections; so all TLS server connections are considered equivalent, and all TLS client connections are considered equivalent. Additionally, HTTPS/TLS is TLS transporting HTTP. There is no difference in the cryptography from stand-alone TLS.

The following cryptographic channels were tested fully tested according to the evaluation activities:

- HTTPS/TLSv1.2 Web Server with or without TLS client authentication – REST API
- TLSv1.2 client with or without TLS client authentication –Zero Reveal Server

The results of the following test cases of cryptographic channel "TLSv1.2 client with or without TLS client authentication – LDAP server" were equivalent to the results of the cryptographic channel "TLSv1.2 client with or without TLS client authentication –Zero Reveal Server" according to the evaluation activities:

- FCS_TLSC_EXT.1.1 Test #1
- FCS_TLSC_EXT.1.1 Test #2
- FCS_TLSC_EXT.1.1 Test #3
- FCS_TLSC_EXT.1.1 Test #4
- FCS_TLSC_EXT.1.1 Test #5.1
- FCS_TLSC_EXT.1.1 Test #5.2
- FCS_TLSC_EXT.1.1 Test #5.3
- FCS_TLSC_EXT.1.1 Test #5.4
- FCS_TLSC_EXT.1.1 Test #5.5
- FCS_TLSC_EXT.1.1 Test #5.6
- FCS_TLSC_EXT.1.1 Test #5.7
- FCS_TLSC_EXT.1.2 Test #1
- FCS_TLSC_EXT.1.2 Test #2
- FCS_TLSC_EXT.1.2 Test #4
- FCS_TLSC_EXT.1.2 Test #5.1
- FCS_TLSC_EXT.1.2 Test #5.2(a)
- FCS_TLSC_EXT.1.2 Test #5.2(b)
- FCS_TLSC_EXT.1.2 Test #5.2(c)
- FCS_TLSC_EXT.1.2 Test #5.3(a)
- FCS_TLSC_EXT.1.2 Test #5.3(b)
- FCS_TLSC_EXT.1.3 Test #1a
- FCS_TLSC_EXT.1.3 Test #1b
- FCS_TLSC_EXT.1.3 Test #1c
- FCS_TLSC_EXT.1.3 Test #2

- FCS_TLSC_EXT.1.3 Test #3
- FCS_TLSC_EXT.1.3 Test #4
- FCS_TLSC_EXT.2.1 Test #1
- FCS_TLSC_EXT.2.1 Test #2
- FCS_TLSC_EXT.3.1 Test #1
- FCS_TLSC_EXT.3.1 Test #2
- FCS_TLSC_EXT.5.1 Test #1

The results of the following test cases of cryptographic channel "HTTPS/TLSv1.2 Web Server with or without TLS client authentication – GUI" were equivalent to the results of the cryptographic channel "HTTPS/TLSv1.2 Web Server with or without TLS client authentication – REST API" according to the evaluation activities:

- FCS_TLSS_EXT.1.1 Test #1
- FCS_TLSS_EXT.1.1 Test #2
- FCS_TLSS_EXT.1.1 Test #4.2
- FCS_TLSS_EXT.1.1 Test #4.3ii
- FCS_TLSS_EXT.1.1 Test #4.4
- FCS_TLSS_EXT.1.2 Test #1
- FCS_TLSS_EXT.1.3 Test #3
- FCS_TLSS_EXT.2.2 Test #1
- FCS_TLSS_EXT.2.2 Test #2
- FCS_TLSS_EXT.2.2 Test #3
- FCS_TLSS_EXT.2.2 Test #4
- FCS_TLSS_EXT.2.2 Test #5
- FCS_TLSS_EXT.2.2 Test #6
- FCS_TLSS_EXT.2.2 Test #7(a)
- FCS_TLSS_EXT.2.2 Test #7(b)
- FCS_TLSS_EXT.2.3 Test #1
- FCS_TLSS_EXT.3.1 Test #1

4 Test Bed Descriptions

4.1 Test Bed
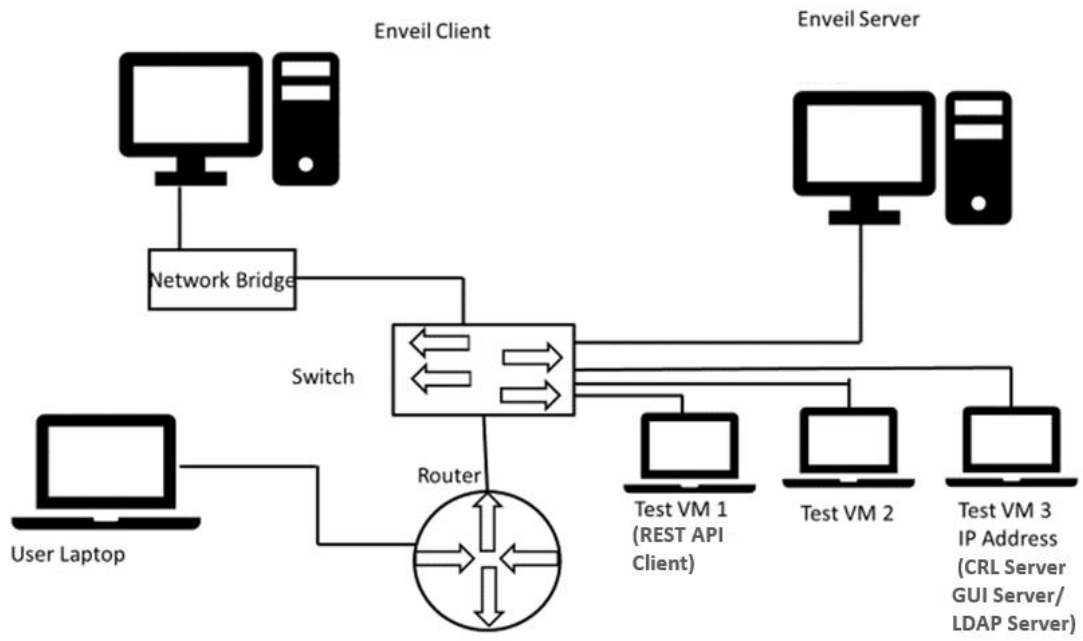
4.2 Test Time and Location

All testing was conducted on the TOE model Rocky Linux 8.7 running software version 4.6.2 and a hot patch of version 4.6.3 fixing the signature algorithm testing (FCS_TLSS_EXT.2.2 Test #3 and FCS_TLSS_EXT.3.1 Test #1) at the Acumen Security offices located in 2400 Research Blvd Suite #395, Rockville, MD 20850. Testing occurred from July 2023 through April 2024.

A regressing testing was also conducted on the TOE model Rocky Linux 8.7 running software version 4.6.3 since a new build was provided, situated at the Acumen Security offices, specifically at 2400 Research Blvd Suite #395, Rockville, MD 20850. The regression testing took place between 4 December, and December 6, 2023, on SFRs 2TLSS, 3 TLSC, 2 TUD and 2 X509 test cases.

Regression Testing was performed on the following test cases:
• TLSS test cases
   o   FCS_TLSS_EXT.1.1 Test 1
   o   FCS_TLSS_EXT.2.2 Test 2

• TLSC test cases
   o   FCS_TLSC_EXT.1.1 Test 2
   o   FCS_TLSC_EXT.1.1 Test 3
   o   FCS_TLSC_EXT.1.3 Test 1a

• TUD test cases
   o   FPT_TUD_EXT.1.1 Test 1
   o   FPT_TUD_EXT.1.2 Test 1

• X509 test cases
   o   FIA_ X509_EXT.1.1 Test #2
   o   FIA_X509_EXT.1.2 Test 1

TOE was in a physically protected, access controlled, designated test lab with no unattended entry/exit ways. At the start of each day, the test bed was verified to ensure that it was not compromised. All evaluation documentation was always kept with the evaluator.

4.3 Configuration Information

The following table provides configuration information about each device in the test environment.

Table 1 Test Configuration Information

Name	OS	Credentials	Function	Protocols	Tools (version)
Enveil Client	Rocky Linux 8.7	admin/thief.escapee.hock	TOE Platform	TLS1.2 SSH (SSH for remote access to the test platform)	acumen-tlsc-pkg curl 7.61.1 OpenSSL 1.1.1k OpenSSH_8.0p1
Enveil Server	Rocky Linux 8.7	admin/thief.escapee.hock	Enveil ZR Server Platform	TLS1.2 SSH (SSH for remote access to the test platform)	acumen-tlsc-pkg curl 7.61.1 OpenSSH_8.0p1 OpenSSL 1.1.1k
Test VM 1	Ubuntu 20.04	acumensec/123TesT321	Test VM with OpenSSL (REST API Client)	TLS1.2	acumen-tlsc-pkg acumen-tlss-pkg curl 7.68.0 OpenSSL 1.1.1f
Test VM 2	Ubuntu 20.04	acumensec/123TesT321	Test VM with OpenSSL	TLS1.2	OpenSSL 3.0.2
Test VM 3	Kali Linux	acumensec/123TesT321	CRL Server/GUI Server/LDAP Server	TLS1.2	OpenSSL 3.0.10
Test VM 4	Kali Linux	acumensec/123TesT321	Test VM with OpenSSL	TLS1.2	OpenSSL 3.0.10
Network Bridge	Ubuntu 20.04	acumensec/123TesT321	Physical Device	SSH	Wireshark 3.2.3
User Laptop	Windows	N/A	Tester Workstation	SSHv2	Wiresharkv3.6.15 XCA v2.4.0

5 Detailed Test Cases (TSS and AGD Activities)

5.1 Mandatory Requirements

5.2 Cryptographic Support (FCS)

5.2.1.1 FCS_CKM.1_EXT.1 Cryptographic Key Generation Services (Applied TD0717)

5.2.1.1.1 FCS_CKM.1.1_EXT.1 TSS

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 reviewed the TSS 6 and ensured that the application does or does not need asymmetric key generation services or that evaluation activities were performed as stated in the selection-based requirements.

The relevant information is found in the following section(s): TOE Summary Specification FCS_CKM_EXT.1.

Upon investigation, the evaluator found that the TSS states that: The TOE implements ECDSA Key Generation, Signature Generation, and Signature Verification as part of TLS trusted channel establishment. NIST curves P-256 and P-384 are supported. The TOE implements RSA Key Generation, Signature Generation and Signature Verification as part of TLS trusted channel establishment. Key sizes of 2048 and 3072-bits and greater are supported.

Key establishment for TLS is performed using Elliptic Curve Diffie-Hellman with NIST curves P-384.

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

Verdict:

PASS.

5.2.1.1.2 FCS_CKM.1.1_EXT.1 AGD

According to the PP, there are no AGD requirements for this SFR.

5.2.1.2 FCS_RBG_EXT.1 Random Bit Generation Services

5.2.1.2.1 FCS_RBG_EXT.1.1 TSS

Objective:

- If "use no DRBG functionality" is selected, the evaluator shall review the TSS to ensure that it needs no random bit generation services.
- If "implement DRBG functionality" is selected, the evaluator shall review the TSS to ensure that additional FCS_RBG_EXT.2 elements are included in the ST.
- If "invoke platform-provided DRBG functionality" is selected, the evaluator shall perform 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 reviewed the TSS 6 and ensured that, if "implement DRBG functionality" is selected, that additional FCS_RBG_EXT.2 elements are included in the ST.
  - The TSS identifies all functions (as described by the SFRs included in the ST) that obtain random numbers from the platform RBG.
  - For each of these functions, the TSS states which platform interface (API) is used to obtain the random numbers.
  - Each of these interfaces corresponds to the acceptable interfaces listed for each platform below.

The relevant information is found in the following section(s): TOE Summary Specification FCS_RBG_EXT.1.

Upon investigation, the evaluator found that the TSS states that: The TOE utilizes a platform based DRBG as its noise source and seeds with a minimum of 256 bits of entropy. This is achieved using the SecureRandom Java class which is configured to use the /dev/random system device.

Additional information related to entropy functionality of the TOE can be reviewed in the Entropy Assessment Report (EAR) provided as an ancillary document.

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

Verdict:
PASS.

5.2.1.2.2   FCS_RBG_EXT.1.1 AGD

According to the PP, there are no AGD requirements for this SFR.

5.2.1.3   FCS_STO_EXT.1 Storage of Credentials

5.2.1.3.1   FCS_STO_EXT.1 TSS

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 reviewed the TSS 6 and ensured that it lists all persistent credentials (secret keys, PKI private keys, or passwords) needed to meet the requirements in the ST.
- The evaluator reviewed the TSS 6 and ensured that it lists for what purpose it is used, and how it is stored.

The relevant information is found in the following section(s): TOE Summary Specification FCS_STO_EXT.1.

Upon investigation, the evaluator found that the TSS states that: TOE implements secure storage of TLS certificates and private keys used as part of establishing the TLS trusted channel with the Enveil ZeroReveal Server, LDAP Server, GUI Users, and REST API Users by encrypting them with AES-CCM and storing them in /etc/enveil/zeroreveal-client/certs. The TOE uses its Bouncy Castle cryptographic library to encrypt/decrypt.

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

5.2.1.3.2 FCS_STO_EXT.1 AGD

According to the PP, there are no AGD requirements for this SFR.

Verdict:

PASS.

5.2.1.4 FCS_TLS_EXT.1 TLS Protocol

5.2.1.4.1 FCS_TLS_EXT.1.1 TSS

According to the PP, there are no TSS requirements for this SFR.

5.2.1.4.2 FCS_TLS_EXT.1.1 AGD

Objective:

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

Evaluator Findings:

The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1".

The relevant information is found in the following section(s):

- "3.4 Configuring ZeroReveal Client" of ZeroReveal Client Guide

Upon investigation, the evaluator found that the AGD activity states that: ZeroReveal Client requires three TLS Certificates:

- o A TLS server certificate to authenticate incoming connections to web-facing services hosted by the ZeroReveal Client. These services can be accessed via a web browser or a REST API client.
- o A TLS client certificate to authenticate outgoing connections to the ZeroReveal Server that must be signed using an algorithm using SHA384 or SHA512 (and not SHA256)
- o A TLS client certificate to authenticate outgoing connections to the LDAP Service that must be signed using an algorithm using SHA384 or SHA512 (and not SHA256)

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

Verdict:

PASS.

5.2.2 User Data Protection (FDP)

5.2.2.1 FDP_DEC_EXT.1 Access to Platform Resources

5.2.2.1.1 FDP_DEC_EXT.1.1 TSS

According to the PP, there are no TSS requirements for this SFR.

5.2.2.1.2 FDP_DEC_EXT.1.1 AGD

Objective:

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

Evaluator Findings:
- The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1"and ensured that the application has access to hardware resources.
- The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured that it is consistent with the selections indicated.
- The evaluator checked the AGD " ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured that it justifies as to why access is required.

The relevant information is found in the following section(s):
- "3.1 System Requirements" of ZeroReveal Client Guide

Upon investigation, the evaluator found that the AGD activity states that: ZeroReveal Client uses the network interface to communicate with users and any ZeroReveal Servers configured with an HTTP connection. Additionally, ZeroReveal Client will use a network connection to the configured LDAP service and to validate certificates if they contain CRL endpoints.

The evaluator also examined 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.

5.2.2.1.3 FDP_DEC_EXT.1.2 TSS

According to the PP, there are no TSS requirements for this SFR.

5.2.2.1.4 FDP_DEC_EXT.1.2 AGD

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.
- 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 checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1"and ensured that the application can access sensitive information repositories.
- The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1"and ensured that it is consistent with the selections indicated.
- The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1"and ensured that it justifies as to why access is required.The relevant information is found in the following section(s):
- 3.4 Configuring ZeroReveal Client of ZeroReveal Client Guide

Upon investigation, the evaluator found that the AGD activity states that: ZeroReveal Client does not provide any access to sensitive information repositories, only the data sources backing the connected ZeroReveal Servers.

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

Verdict:

PASS.

5.2.2.2   FDP_NET_EXT.1 Network Communications

5.2.2.2.1   FDP_NET_EXT.1.1

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.2.2.3   FDP_DAR_EXT.1 Encryption Of Sensitive Application Data

5.2.2.3.1   FDP_DAR_EXT.1.1 TSS

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

Evaluator Findings:
- The evaluator reviewed the TSS 6 and ensured that it describes the sensitive data processed by the application.
- The evaluator reviewed the TSS 6 and ensured that the activities cover all of the sensitive data identified in the TSS.
- The evaluator reviewed the TSS 6 and ensured that, "not store in sensitive data" is selected, it describes how sensitive data cannot be written to non-volatile memory.
- The evaluator reviewed the TSS 6 and ensured that this is consistent with the filesystem test below.

The relevant information is found in the following section(s): TOE Summary Specification FDP_DAR_EXT.1.

Upon investigation, the evaluator found that the TSS states that: The TOE protects application log files and configuration data (stored in /var/log/enveil/zeroreveal-client/client.log, and /var/log/enveil/zeroreveal-client/stacks.log, and /etc/enveil/zeroreveal-client/client.conf) using Linux filesystem encryption. The log files are considered sensitive data because the files are very verbose and include certificate information. The configuration file includes passwords that enable the TOE to decrypt the files using the Linux file system.

The TOE implements secure storage of TLS certificates and private keys (stored in /etc/enveil/zeroreveal-client/certs) in accordance with FCS_STO_EXT.1 which uses the TOE's Bouncy Castle cryptographic library to encrypt with AES-CCM. The TLS certificates and private keys are encrypted again by the Linux platform provided encryption/decryption functions.

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

Verdict:
PASS.

5.2.2.3.2   FDP_DAR_EXT.1.1 AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.2.3 Security Management (FMT)

5.2.3.1   FMT_MEC_EXT.1 Supported Configuration Mechanism

5.2.3.1.1 FMT_MEC_EXT.1.1 TSS

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 AGD in response to an SFR.

- 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 reviewed the TSS 6 and ensured that it 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.

- The evaluator reviewed the TSS 6 and ensured that, at a minimum, it list settings related to any SFRs and any settings that are mandated in the AGD in response to an SFR.

- The evaluator reviewed the TSS 6 and ensured that, 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 TSS identifies those options, as well as indicates where the encrypted representation of these options is stored.

The relevant information is found in the following section(s): TOE Summary Specification FMT_MEC_EXT.1.

Upon investigation, the evaluator found that the TSS states that: Configuration files (modifiable by a text editor) are used to manage TOE configuration. Non-functional configuration file templates are put in place by the installer package. Global configuration options are stored in the /etc/enveil/zeroreveal-client directory.

The TOE invokes the mechanisms recommended by the platform vendor for storing and setting configuration options.

The following parameters are required to be configured: Refer to Appendix.

Also, the ST does not claim FDP_PRT_EXT.1.

The evaluator reviewed the AGD and section "3.4 Configuring ZeroReveal Client" states that : Ensure that all TLS key stores and TLS trust stores are stored in /etc/enveil/zeroreveal-client/certs/ and are readable only by the enveil user.
Make sure that the following Common Criteria specific options in client.conf are set with the following constraints:

enveil.security.tls.keystore.path
    Path to the key store on ZeroReveal Server's local disk.
enveil.security.tls.keystore.type
    Type of the key store (possible options are jks, pkcs12, or bcfks).
enveil.security.tls.keystore.password
    The key store's password.
enveil.security.tls.truststore.path
    Path to the trust store on ZeroReveal Server's local disk.
enveil.security.tls.keystore.type
    Type of the key store (possible options are jks or bcfks).
enveil.security.tls.truststore.password
    The trust store's password.

If the certificate keys are generated using Elliptic Curve Cryptography, ensure that the curve used is either secp256r1 or secp384r1. If RSA keys are used, they must be 2048, 3072 bits, or 4096 bits.

Ensure that all TLS key stores and TLS trust stores are stored in /etc/enveil/zeroreveal-client/certs/ and are readable only by the enveil user.

enveil.common.niap.enforce

    (boolean) Enforces that the server is configured to meet the NIAP requirements.

    Must be set to true.

enveil.client.auth.mechanisms

    Comma-separated list of authentication mechanisms to use.

    Must be set to [certificate].

enveil.client.auth.require.user.cert

    (boolean) Whether to require users to present valid TLS client certificates.

    Must be set to true.

enveil.client.auth.certificate.user.source.mechanisms

    (string) A comma-separated list of user stores for use with certificate authentication.

    Must be set to [ldap].

enveil.client.auth.certificate.ldap.ssl.enabled

    (boolean) Whether to connect to the LDAP server under TLS for certificate enveil.client.auth.

    Must be set to true.

enveil.client.auth.certificate.ldap.connect.with.sasl.external

    (boolean) Whether to authenticate to the LDAP server using a TLS client certificate or not for certificate auth.

    Must be set to true.

enveil.client.gateway.specification.dir

    (path) Path to a directory containing specifications for ZeroReveal Servers to connect to. Each ZeroReveal Server is represented by a separate properties file.

    Must be set to /etc/enveil/zeroreveal-client/gateways.

enveil.security.tls.conscrypt.aes.enabled

    (boolean) true enables the use of native AES ciphers from a bundled BoringSSL implementation. false will disable the native ciphers and use default Java implementation.

    Must be set to false.

enveil.security.tls.keystore.check

    (boolean) Validates the key store on startup.

    Must be set to true.

enveil.security.tls.strict

(string) If true, requires TLSv1.2 and an AES-256 cipher suite for all connections. (Note that the elliptic curve used with these cipher suites for key establishment is only secp384r1. If false, accepts any valid TLS protocol and cipher suite available in the local Java installation.

    Must be set to true.

enveil.security.tls.client.certificate.check

    (boolean) Whether to check the validity of a certificate presented by any TLS client (currently only ZeroReveal Client).

    Must be set to true.

enveil.security.random.blockingDevice

    (boolean) Whether to use a blocking device for random number generation. That is, wait for enough entropy to be available before generating random numbers.

    Must be set to true.

enveil.security.tls.niap.signature.algorithms

    (boolean) Only used NIAP-approved signature algorithms.

    Must be set to true.

enveil.security.cert.revocation.check.mode

> (string) Whether to check for certificate revocation using any provided CRL endpoint. Defaults to NONE. Must be set to "HARD_FAIL".

ZeroReveal Client automatically restricts all TLS connections to TLS version 1.2, denying all other TLS versions. No further configuration is required to configure the cryptographic engine beyond the parameters described above.

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

Verdict:

PASS.

5.2.3.1.2   FMT_MEC_EXT.1.1 AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.2.3.2   FMT_CFG_EXT.1 Secure by Default Configuration

5.2.3.2.1   FMT_CFG_EXT.1.1 TSS

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 reviewed the TSS 6 and ensured that, if the application requires any type of credentials and if the application installs with default credentials.

The relevant information is found in the following section(s): TOE Summary Specification FMT_CFG_EXT.1.

Upon investigation, the evaluator found that the TSS states that: The TOE is not installed with default credentials. The TOE installer package makes sure all configuration and data directories are configured with appropriate permissions to restrict against modification by unprivileged users.

Once the TOE has been installed, the following configuration steps must be completed:

• Set up TLS for the TOE.

• Configure the TOE's LDAP interaction.

• Assign the TOE client permissions in LDAP.

• Configure at least one ZeroReveal Compute Fabric Server connection.

The TOE does not provide any functionality until an administrator provides modified (non-default) configuration files.

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

Verdict:

PASS.

5.2.3.2.2   FMT_CFG_EXT.1.1 AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.2.3.2.3   FMT_CFG_EXT.1.2

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.2.3.3 FMT_SMF.1 Specification of Management Functions

5.2.3.3.1 FMT_SMF.1.1 TSS

According to the PP, there are no AA requirements for TSS activity for this SFR.

5.2.3.3.2 FMT_SMF.1.1 AGD

Objective:

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

Evaluator Findings:

The evaluator checked the AGD and ensured that every management function mandated by the PP is described in the AGD and that the description contains the information required to perform the management duties associated with the management function.

- " 1.1 Targets of Evaluation and Scope"

Upon investigation, the evaluator found that the AGD activity states that: For both ZeroReveal Client and ZeroReveal Server an administrator (not necessarily the same person) manages the TOE via the configuration files, there are no management interfaces other than that.

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

Verdict:

PASS.

5.2.4 Privacy (FPR)

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

5.2.4.1.1 FPR_ANO_EXT.1.1 TSS

Objective:

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

Evaluator Findings:

The evaluator reviewed the TSS 6 and ensured that it identifies functionality in the application where PII can be transmitted.

The relevant information is found in the following section(s): TOE Summary Specification FPR_ANO_EXT.1.

Upon investigation, the evaluator found that the TSS states that: The TOE does not collect or transmit PII over a network.

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

Verdict:

PASS.

5.2.4.1.2 FPR_ANO_EXT.1.1 AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.2.5 Protection of the TSF (FPT)

5.2.5.1 FPT_API_EXT.1 Use of Supported Services and APIs

5.2.5.1.1   FPT_API_EXT.1.1 TSS

Objective:

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

Evaluator Findings:

The evaluator reviewed the TSS 6 and ensured that it lists the platform APIs used in the application.

The relevant information is found in the following section(s): TOE Summary Specification FPT_API_EXT.1.

Upon investigation, the evaluator found that the TSS states that: Enveil only uses public APIs in the TOE. The TOE uses the Linux APIs identified in Appendix A of the ST.

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

Verdict:

PASS.

5.2.5.1.2   FPT_API_EXT.1.1 AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.2.5.2   FPT_AEX_EXT.1 Anti-Exploitation Capabilities

5.2.5.2.1   FPT_AEX_EXT.1.1 TSS (Applied 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 reviewed the TSS 6 and ensured that it describes the compiler flags used to enable ASLR when the application is compiled.

The relevant information is found in the following section(s): TOE Summary Specification FPT_AEX_EXT.1

Upon investigation, the evaluator found that the TSS states that: The main TOE application code is written in Java which places calls out to native C/C++ binaries. The Java binaries rely on the JRE for memory and stack protection, which are compiled into the JRE used in the OE by the JRE vendor.

The two native code libraries in the TOE: SEAL and GMP.

GMP and SEAL are compiled using GCC with the required compiler flags for ASLR (GCC CFLAG –fPIC, "Generate position-independent code") and stack protection (-fstackprotector-all).

The memory protections for the GMP and SEAL native code portion were verified through static analysis. The TOE allocates memory regions with write and execute permissions for OpenJDK Java runtime performing just-in-time compilation.

The TOE installs data and library files to /usr/local/enveil/* and configuration files to /etc/enveil/*. By default, the installed directories containing user-modifiable files do not have executables in them.

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

Verdict:

PASS.

5.2.5.2.2   FPT_AEX_EXT.1.1 AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.2.5.2.3 FPT_AEX_EXT.1.2

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.2.5.2.4 FPT_AEX_EXT.1.3

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.2.5.2.5 FPT_AEX_EXT.1.4

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.2.5.2.6 FPT_AEX_EXT.1.5 TSS (Applied 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 evaluator reviewed the TSS 6 and ensured that it the stack-based buffer overflow compiler flags.

The relevant information is found in the following section(s): TOE Summary Specification FPT_AEX_EXT.1

Upon investigation, the evaluator found that the TSS states that: GMP and SEAL are compiled using GCC with the required compiler flags for ASLR (GCC CFLAG –fPIC, "Generate position-independent code") and stack protection (-fstackprotector-all).

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

Verdict:

PASS.

5.2.5.2.7 FPT_AEX_EXT.1.5 AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.2.5.3 FPT_IDV_EXT.1 Software Identification and Versions

5.2.5.3.1 FPT_IDV_EXT.1.1 TSS

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 reviewed the TSS 6 and ensured that if "other version information" is selected, the TSS contains an explanation of the versioning methodology.

The relevant information is found in the following section(s): TOE Summary Specification FPT_IDV_EXT.1.

Upon investigation, the evaluator found that the TSS states that: The TOE is versioned with version information published in the installation guidance. The TOE versioning methodology is Major Version.Minor Version.Patch Level.

The evaluated TOE's version is 4.6.3.

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

Verdict:

PASS.

5.2.5.3.2   FPT_IDV_EXT.1.1 AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.2.5.4   FPT_LIB_EXT.1 Use of Third-Party Libraries

5.2.5.4.1   FPT_LIB_EXT.1.1

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.2.5.5   FPT_TUD_EXT.1 Integrity for Installation and Update

5.2.5.5.1   FPT_TUD_EXT.1.1 TSS

According to the PP, there are no AA requirements for TSS activity for this SFR.

5.2.5.5.2   FPT_TUD_EXT.1.1 AGD

Objective:

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

Evaluator Findings:

The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured that it includes a description of how updates are performed.

The relevant information is found in the following section(s):

- 3.6 Updating ZeroReveal Client of ZeroReveal Client Guide

Upon investigation, the evaluator found that the AGD activity states that: ZeroReveal Client may only be updated using the package manager, by running the following command:

```
bash$ yum update enveil-zeroreveal-client
```

This will display whether or not an update is available and if so, ask whether to apply the update. ZeroReveal Client may only be updated using the package manager. The package manager will automatically reject any update that is either not signed or signed with the wrong key

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

Verdict:

PASS.

5.2.5.5.3   FPT_TUD_EXT.1.2 TSS

According to the PP, there are no AA requirements for TSS activity for this SFR.

5.2.5.5.4   FPT_TUD_EXT.1.2 AGD

Objective:

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

Evaluator Findings:

The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured that it includes a description of how to query the current version of the application.

The relevant information is found in the following section(s):

3.5 Determining the Installed Version of ZeroReveal Client

Upon investigation, the evaluator found that the AGD activity states that: To examine which version of ZeroReveal Client is installed run the following command:

```
bash$ yum info enveil-zeroreveal-client
```

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

Verdict:

PASS.

5.2.5.5.5   FPT_TUD_EXT.1.3

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.2.5.5.6   FPT_TUD_EXT.1.4 TSS

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 AGD) describes how candidate updates are obtained.

Evaluator Findings:

- The evaluator reviewed the TSS 6 and ensured that it 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 reviewed the TSS 6 and ensured that it describes how candidate updates are obtained.

The relevant information is found in the following section(s): TOE Summary Specification FPT_TUD_EXT.1.

Upon investigation, the evaluator found that the TSS states that: The TOE supports packages running on Red Hat and Red Hat derivatives in RPM format. Official Enveil RPMs are signed using Enveil's private signing key. When using Yum to install Enveil TOE packages, the GPG signatures on the RPM files will automatically be checked. If they are missing a signature or signed with the wrong GPG key, then an error indicating that the GPG keys for the repository do not match the package will be displayed and the install will automatically abort. These checks are also run during the installation of every update.

The TOE records its version in the RPM package file. An administrator can determine the current version by running the command yum info zeroreveal-client.

The update/install packages include the required information so that the package manager will perform removal and deletion of all traces of the application when an uninstall command is issued through that package manager.

The TOE is updated using the platform package manager. When Enveil developers finish a new version of any component, they sign then upload it to the package repositories, which make it available to users. Updates are initiated by users via the package manager; the TOE will never download, modify, replace or update its own binary code.

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

Verdict:

PASS.

5.2.5.5.7   FPT_TUD_EXT.1.4 AGD

According to the PP, there are no AA requirements for AGD activities for this SFR.

5.2.5.5.8   FPT_TUD_EXT.1.5 TSS

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.

Evaluator Findings:

- The evaluator reviewed the TSS 6 and ensured that it identifies how the application is distributed.

- The evaluator reviewed the TSS 6 and ensured that, if "with the platform" is selected, that TOE software is included as part of the platform OS.

The relevant information is found in the following section(s): TOE Summary Specification FPT_TUD_EXT.1.

Upon investigation, the evaluator found that the TSS states that: Enveil will publish Yum repositories for updates and patches to the TOE. The TOE relies on Yum to periodically poll the repositories for updates and notify the user. The TOE does not check for or apply updates on its own.

The TOE supports packages running on Red Hat and Red Hat derivatives in RPM format. Official Enveil RPMs are signed using Enveil's private signing key. When using Yum to install Enveil TOE packages, the GPG signatures on the RPM files will automatically be checked. If they are missing a signature or signed with the wrong GPG key, then an error indicating that the GPG keys for the repository do not match the package will be displayed and the install will automatically abort. These checks are also run during the installation of every update.

Verdict:

PASS.

5.2.5.5.9   FPT_TUD_EXT.1.5 AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.2.6 Trusted Path/Channels (FTP)

5.2.6.1   FTP_DIT_EXT.1 Protection of Data in Transit

5.2.6.1.1   FTP_DIT_EXT.1.1 TSS

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 reviewed the TSS 6 and ensured that it contains the calls to the platform that TOE is leveraging to invoke the functionality.

The relevant information is found in the following section(s): TOE Summary Specification FTP_DIT_EXT.1.

Upon investigation, the evaluator found that the TSS states that: Users communicate with the TOE through REST interfaces via TLS and GUI interfaces via mutually authenticated HTTPS/TLS.

Communication between the TOE and a ZeroReveal Compute Fabric Server is via REST over HTTPS/TLS.

The TOE communicates with an authentication server using Lightweight Directory Access Protocol (LDAP) secured with TLS.

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

Verdict:

PASS.

5.2.6.1.2   FTP_DIT_EXT.1.1 AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.3   Optional Requirements

5.3.1 Cryptographic Support (FCS)

5.3.1.1 FCS_CKM.1/SK Cryptographic Symmetric Key Generation

5.3.1.1.1 FCS_CKM.1.1/SK TSS

Objective:

- The evaluator shall review the TSS to determine that it describes how the functionality described by FCS_RBG_EXT.1 is invoked.
- If the application is relying on random bit generation from the host platform, the evaluator shall verify the TSS includes the name/manufacturer of the external RBG and describes the function call and parameters used when calling the external DRBG function.
- If different external RBGs are used for different platforms, the evaluator shall verify the TSS identifies each RBG for each platform.
- Also, the evaluator shall verify the TSS includes a short description of the vendor's assumption for the amount of entropy seeding the external DRBG. The evaluator uses the description of the RBG functionality in FCS_RBG_EXT or documentation available for the operational environment to determine that the key size being requested is identical to the key size and mode to be used for the encryption/decryption of the user data.

Evaluator Findings:

- The evaluator reviewed the TSS 6 and ensured that it describes how the functionality described by FCS_RBG_EXT.1 is invoked.
- The evaluator reviewed the TSS 6 and ensured that, if the application is relying on random bit generation from the host platform, the TSS includes the name/manufacturer of the external RBG and describes the function call and parameters used when calling the external DRBG function.
- The evaluator reviewed the TSS 6 and ensured that, if different external RBGs are used for different platforms, the TSS identifies each RBG for each platform.
- The evaluator reviewed the TSS 6 and ensured that it includes a short description of the vendor's assumption for the amount of entropy seeding the external DRBG.

The relevant information is found in the following section(s): TOE Summary Specification FCS_CKM.1/SK and FCS_RBG_EXT.1.

Upon investigation, the evaluator found that the TSS states that: The TOE implements HMAC_DRBG functionality to generate random bits for use in the rest of the cryptographic functions. The TOE utilizes a platform based DRBG as its noise source and seeds with a minimum of 256 bits of entropy. This is achieved using the SecureRandom Java class which is configured to use the /dev/random system device.

The TOE generates symmetric AES 256-bit keys for use in AES-GCM as part of TLS and for use in AES-CCM for protection of stored credentials.

Refer to the ancillary Entropy Assessment Report for information about entropy details.

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

Verdict:

PASS.

5.3.1.1.1 FCS_CKM.1.1/SK AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.4 Selection-Based Requirements

5.4.1 Cryptographic Support (FCS)

5.4.1.1 FCS_CKM.1/AK Cryptographic Asymmetric Key Generation

5.4.1.1.1 FCS_CKM.1.1/AK TSS

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," the evaluator shall examine the TSS to verify that it describes how the key generation functionality is invoked.

Evaluator Findings:
- The evaluator reviewed the TSS 6 and ensured that it identifies the key sizes supported by the TOE.
- The evaluator reviewed the TSS 6 and ensured that, if the ST specifies more than one scheme, the TSS identifies the usage for each scheme.
- The evaluator reviewed the TSS 6 and ensured that, if the application "invokes platform-provided functionality for asymmetric key generation," the TSS describes how the key generation functionality is invoked.

The relevant information is found in the following section(s): TOE Summary Specification FCS_CKM.1/AK.

Upon investigation, the evaluator found that the TSS states that: The TOE implements ECDSA Key Generation, Signature Generation, and Signature Verification as part of TLS trusted channel establishment. NIST curves P-256 and P-384 are supported.

The TOE implements RSA Key Generation, Signature Generation and Signature Verification as part of TLS trusted channel establishment. Key sizes of 2048 and 3072-bits and greater are supported.

The evaluator examined the SFR in the Security Target and determined that invoke platform-provided functionality is not selected.

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

Verdict:
PASS.

5.4.1.1.2 FCS_CKM.1.1/AK AGD

The evaluator shall verify that the AGD 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 checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured 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.

The relevant information is found in the following section(s):

- "5.4 Supported key types" of ZeroReveal Client Guide

Upon investigation, the evaluator found that the AGD activity states that:

RSA Keys

RSA keys of size 2048, 3072, and 4096 are supported and can be generated with invocations of `csr-utility` like the following:

```
bash$ csr-utility rsa <key-size> # where <key-bits> is 2048, 3072, or 4096
```

Elliptic Curve Keys

NIST Curve P-256 keys are supported and can be generated with invocations of `csr-utility` like the following:

```
bash$ csr-utility ec secp256r1
```

NIST Curve P-384 keys are supported and can be generated with invocations of `csr-utility` like the following:

```
bash$ csr-utility ec secp384r1
```

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

Verdict:

PASS.

5.4.1.2   FCS_CKM.2 Cryptographic Key Establishment (Applied TD0717)

5.4.1.2.1   FCS_CKM.2.1 TSS

Objective:

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

Evaluator Findings:

- The evaluator reviewed the TSS 6 and ensured that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1/AK.
- The evaluator reviewed the TSS 6 and ensured that, if the ST specifies more than one scheme, it identifies the usage for each scheme.

The relevant information is found in the following section(s): TOE Summary Specification FCS_CKM.2.1

Upon investigation, the evaluator found that the TSS states that: Key establishment for TLS is performed using Elliptic Curve Diffie-Hellman with NIST curves P-384.

Verdict:

PASS.

5.4.1.2.2   FCS_CKM.2.1 AGD

Objective:

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

Evaluator Findings:

The evaluator checked the AGD and ensured that it instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).The relevant information is found in the following section(s):

- "3.4 Configuring ZeroReveal Client" of ZeroReveal Client Guide

Upon investigation, the evaluator found that the AGD activity states that: Make sure that the following Common Criteria specific options in client.conf are set with the following constraints:

enveil.security.tls.strict

(string) If true, requires TLSv1.2 and one of the following cipher suites for all connections:

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384.

(Note that the elliptic curve used with these cipher suites for key establishment is only secp384r1.)

If false, accepts any valid TLS protocol and cipher suite available in the local Java installation. (Note that the elliptic curve used with these cipher suites for key establishment is only $secp384r1$.)

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

Verdict:

PASS.

5.4.1.3   FCS_COP.1/Sig Cryptographic Operation – Signing

5.4.1.3.1   FCS_COP.1.1/Sig

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

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

5.4.1.4.1   FCS_COP.1.1/SKC TSS

According to the PP, there are no AA requirements for TSS activity for this SFR.

5.4.1.4.2   FCS_COP.1.1/SKC AGD

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 checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured that it provides documentation to determine that any configuration that is required to be done to configure the functionality for the required modes and key sizes is present.

The relevant information is found in the following section(s):

- "3.4 Configuring ZeroReveal Client" of ZeroReveal Client
- "5.5 Creating a bcfks from a p12"
- "5.5.1 On a machine with ZeroReveal Client installed"

Upon investigation, the evaluator found that the AGD activity states that: Make sure that the following Common Criteria specific options in client.conf are set with the following constraints:

enveil.security.tls.strict

(string) If true, requires TLSv1.2 and one of the following cipher suites for all connections:

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384.(Note that the elliptic curve used with these cipher suites for key establishment is only $secp384r1$.) If false, accepts any valid TLS protocol and cipher suite available in the local Java installation. Must be set to true.

For the configuration to be Common Criteria compliant, the keystores must be in the bcfks format. See Creating a bcfks from a .p12 for instructions on converting keystores to the bcfks format. The steps below ensure that the Bouncy Castle cryptographic library uses AES-CCM for protection of stored credentials.

NOTE: Be sure to use the same password for the new key store as was used for the existing key store otherwise keytool may produce a corrupted key store

To convert keystore-in.p12 to keystore-out.bcfks on a machine with ZeroReveal Client installed, run the following command. You will be prompted for the password to decrypt keystore-in.p12 and a new password for keystore-out.bcfks, be sure to use the same password for both.

bash$ keytool -providerclass org.bouncycastle.jcajce.provider.BouncyCastleFipsProvider \ -providerpath /usr/local/enveil/zeroreveal-client/bin/bouncy-castle-fips.jar \ -importkeystore -srcstoretype bcpkcs12 - deststoretype bcfks \ -srckeystore keystore-in.p12 -destkeystore keystore-out.bcfks

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

Verdict:
PASS.

5.4.1.5   FCS_COP.1/Hash Cryptographic Operation - Hashing

5.4.1.5.1   FCS_COP.1.1/Hash TSS

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 reviewed the TSS 6 and ensured that the digital signature verification function is documented in the TSS.

The relevant information is found in the following section(s): TOE Summary Specification FCS_COP.1/Hash

Upon investigation, the evaluator found that the TSS states that: The TOE performs hashing and HMAC using:

● SHA-256, using a 512-bit block size and 256-bit message digest size as part of digital signatures.

● SHA2-384, using a 1024-bit block size and 384-bit message digest size as part of TLS and digital signatures.

● SHA2-512, using a 1024-bit block size and 512-bit message digest size as part of the authentication function used in key store and certificate formatting, and as the underlying DRBG function.

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

Verdict:
PASS.

5.4.1.5.2   FCS_COP.1.1/Hash AGD
According to the PP, there are no AA requirements for AGD activity for this SFR.

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

5.4.1.6.1   FCS_COP.1.1 /KeyedHash
According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.4.1.7   FCS_HTTPS_EXT.1/Client HTTPS Protocol

5.4.1.7.1   FCS_HTTPS_EXT.1.1/Client TSS

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 reviewed the TSS 6 and ensured that it provides enough detail to explain how the implementation complies with RFC 2818.

The relevant information is found in the following section(s): TOE Summary Specification FCS_HTTPS_EXT.1/Client.

Upon investigation, the evaluator found that the TSS states that: The TOE acts as an HTTPS Client when communicating to the ZeroReveal Servers and the LDAP Servers.

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. HTTPS is implemented using TLS 1.2 (RFC 5246).

The TOE's REST API interface to ZeroReveal Servers and interface to LDAP Servers rejects a connection when a server's certificate is invalid. If a Server's certificate is deemed invalid, the TOE will notify the user by writing into the var/log/enveil/zeroreveal-client/client.log log file.

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

Verdict:

PASS.

5.4.1.7.2   FCS_HTTPS_EXT.1.1/Client AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.4.1.7.3   FCS_HTTPS_EXT.1.2/Client

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.4.1.7.4   FCS_HTTPS_EXT.1.3/Client

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.4.1.8   FCS_HTTPS_EXT.1/Server HTTPS Protocol

5.4.1.8.1   FCS_HTTPS_EXT.1.1/Server TSS

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 reviewed the TSS 6 and ensured that it provides enough detail to explain how the implementation complies with RFC 2818.

The relevant information is found in the following section(s): TOE Summary Specification FCS_HTTPS_EXT.1/Server

Upon investigation, the evaluator found that the TSS states that: The TOE acts as an HTTPS Server when receiving REST API and GUI requests from clients.

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. HTTPS is implemented using TLS 1.2 (RFC 5246).

The TOE REST interface and GUI interface reject a connection if a User Workstation's certificate is invalid (mutual authentication) based on an administrator configurable parameter and notify the user.

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

Verdict:

PASS.

5.4.1.8.2 FCS_HTTPS_EXT.1.1/Server AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.4.1.8.3 FCS_HTTPS_EXT.1.2/Server

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.4.1.8.4 FCS_HTTPS_EXT.1.3/Server

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.4.1.9   FCS_HTTPS_EXT.2 HTTPS Protocol with Mutual Authentication

5.4.1.9.1 FCS_HTTPS_EXT.2.1

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.4.1.10   FCS_RBG_EXT.2 Random Bit Generation from Application

5.4.1.10.1 FCS_RBG_EXT.2.1 TSS

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.4.1.10.2 FCS_RBG_EXT.2.2 TSS

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:

The evaluator reviewed the TSS 6 and ensured that it provides documentation in accordance with Appendix C - Entropy Documentation and Assessment and the Clarification to the Entropy Documentation and Assessment Annex.

The relevant information is found in the following section(s): TOE Summary Specification FCS_RBG_EXT.2

Upon investigation, the evaluator found that the TSS states that: The TOE implements HMAC_DRBG functionality to generate random bits for use in the rest of the cryptographic functions. The TOE utilizes a platform based DRBG as its noise source and seeds with a minimum of 256 bits of entropy. This is achieved using the SecureRandom Java class which is configured to use the /dev/random system device.

Additional information related to entropy functionality of the TOE can be reviewed in the Entropy Assessment Report (EAR) provided as an ancillary document.

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

Verdict:

PASS.

5.4.1.10.3 FCS_RBG_EXT.2.2 AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.4.1.11 *FCS_TLS_EXT.1 TLS Protocol*

5.4.1.11.1 FCS_TLS_EXT.1.1 TLS Protocol TSS

According to the PP, there are no AA requirements for TSS activity for this SFR.

5.4.1.11.2 FCS_TLS_EXT.1.1 TLS Protocol AGD

According to the PP, there are no AA requirements for AGD activity for this SFR

5.4.1.12 *FCS_TLSC_EXT.1 TLS Client Protocol*

5.4.1.12.1 FCS_TLSC_EXT.1.1 TSS

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 reviewed the TSS to ensure that the cipher suites supported are specified.
- The evaluator reviewed the TSS to ensure that the cipher suites specified include those listed for this component.

The relevant information is found in the following section(s): TOE Summary Specification FCS_TLSC_EXT.1

Upon investigation, the evaluator found that the TSS states that: When acting as a TLS client, the TOE implements TLSv1.2 and rejects all older TLS and SSL versions, and supports the following cipher suites:

• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

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

Verdict:

PASS.

5.4.1.12.2 FCS_TLSC_EXT.1.1 AGD

Objective:

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

Evaluator Findings:

The evaluator checked the AGD and ensured that it contains instructions on configuring the product so that TLS conforms to the description in the TSS.

The relevant information is found in the following section(s):

- 3.4 Configuring ZeroReveal Client of ZeroReveal Client Guide

Upon investigation, the evaluator found that the AGD activity states that: ZeroReveal Client automatically restricts all TLS connections to TLS version 1.2, denying all other TLS versions. No further configuration is required to configure the cryptographic engine beyond the steps in this guide. Note: The combination of the settings for enveil.client.auth.mechanisms and enveil.client.auth.require.user.cert ensures that ZeroReveal Client will only accept connections with mutual TLS.

Make sure that the following Common Criteria specific options in client.conf are set with the following

constraints:

enveil.security.tls.strict

(string) If true, requires TLSv1.2 and one of the following cipher suites for all connections: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384. (Note that the elliptic curve used with these cipher suites for key establishment is only secp384r1.) If false, accepts any valid TLS protocol and cipher suite available in the local Java installation. Must be set to true.

enveil.client.auth.mechanisms

Comma-separated list of authentication mechanisms to use. Must be set to [certificate].

enveil.client.auth.require.user.cert

(boolean) Whether to require users to present valid TLS client certificates. Must be set to true.

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

Verdict:

PASS.

5.4.1.12.3 FCS_TLSC_EXT.1.2 TSS

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.

- 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 reviewed the TSS to ensure that it describes the client's method of establishing all reference identifiers from the application-configured reference identifier, including which types of reference identifiers are supported (e.g. Common Name, DNS Name, URI Name, Service Name, or other application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported.

- The evaluator reviewed the TSS to ensure that it identifies whether and the manner in which certificate pinning is supported or used by the product.

The relevant information is found in the following section(s): TOE Summary Specification FCS_TLSC_EXT.1.

Upon investigation, the evaluator found that the TSS states that: The TOE acts as a TLS client when establishing connection to LDAP directory for authentication and when establishing connection to ZeroReveal Compute Fabric Server for operation requests and responses.

When acting as a TLS client, the TOE supports mutual authentication using X.509v3 certificates. The TOE's certificate must contain the hostname or the IP address of the TOE's host machine as a Subject Alternative Name (SAN). The TOE validates the presented identifier in accordance with RFC 6125, and permits the reference identifier to be the CN, DN, or SAN-DNS. Where present, the SAN-DNS identifier supersedes the DN or CN values. Wildcards are supported, only in the leftmost label of the DNS identifier (i.e., "*.example.server.com" but not "example.*.server.com").

The TOE does not support certificate pinning.

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

Verdict:

PASS.

5.4.1.12.4 FCS_TLSC_EXT.1.2 AGD

Objective:

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

Evaluator Findings:

The evaluator checked the AGD" ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured that it includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS.

The relevant information is found in the following section(s):

- 3.4 Configuring ZeroReveal Client of ZeroReveal Client Guide

- 4.3 ZeroReveal Client LDAP Configuration

- 6.2.1.2 Configure LDAP searches

- 6.3.1 Adding a Gateway Specification File

Upon investigation, the evaluator found that the AGD activity states that:

For the configuration to be Common Criteria compliant, the key stores must be in the bcfks format. See *Creating a bcfks from a p12* for instructions on converting key stores to the bcfks format. Furthermore, certificates must contain the hostname or the IP address of the ZeroReveal Client's host machine as a Subject Alternative Name. When creating your certificate, make sure it contains the hostname or IP address of the ZeroReveal Client host machine as a Subject Alternative Name (SAN).

In addition to the settings in Configuring ZeroReveal Client, the following must occur as well:
For a Common Criteria compliant configuration, ZeroReveal Client can use an LDAP server to authenticate users. The LDAP server must be configured to use TLS and the ZeroReveal Client must contain a TLS client certificate/key pair whose certificate is trusted by the LDAP server. The certificate that a user presents to the ZeroReveal Client must be trusted by the ZeroReveal Client. Moreover, the subject alternative name on the user's certificate can be forwarded to the LDAP server for authorization purposes. The distinguished name of the ZeroReveal Client's certificate can be configured in the LDAP server as an administrator in order to lookup the user ID taken from the subject alternative name of the user's certificate. Some additional details are provided in the LDAP Configuration.

ZeroReveal Client how to verify a user's identity and pull their permissions from LDAP. When a user connects to ZeroReveal Client, it does two LDAP searches: the first looks up their Distinguished Name (DN) and verifies their certificate; the second looks up the set of groups the user is associated with. The user's permissions are the union of those we find in the Enveil Permission Attribute (explained below) on the user or any of their groups.

First, set these properties:
enveil.client.auth.ldap.search.base
The search root in the LDAP tree. Example: dc=example,dc=org

enveil.client.auth.ldap.permission.attribute.name
The name of the Enveil Permission Attribute. Example: enveilPermission. Defaults to enveilPlatformPermission.

Next, you must configure how ZeroReveal Client will look up users in LDAP. This is done using the Subject Alternative Name (SAN) on the certificate. For example, uid=Bob,ou=Users,dc=example,dc=org is a valid SAN.

The ZeroReveal Client will then perform DN lookup based on the SAN, using enveil.client.auth.ldap.user.dn.full.lookup.filter. This is a templated LDAP search filter3 that allows fine-grained control over the LDAP search for a given SAN. It must contain "{0}"; when a user attempts to connect to ZeroReveal Client, any "{0}"'s will be replaced with their user ID (uid) and the resulting LDAP search filter will be run.

For example, if client.conf contains:
enveil.client.auth.ldap.search.base=dc=example,dc=org
enveil.client.auth.ldap.user.dn.full.lookup.filter=(& (objectClass=person) (uid={0}))

Then if a user Bob attempts to log in to ZeroReveal Client, we will run an LDAP search with base dc=example,dc=org and filter: (& (objectClass=person) (uid=Bob))

This search must return exactly one object to be considered successful; that object's DN is then treated as Bob's DN. The provided certificate is also verified as the certificate associated with this DN. If this object has an Enveil Permission Attribute, they will be Bob's initial permissions for ZeroReveal Client.

A gateway specification file is a properties file that configures how ZeroReveal Client connects to ZeroReveal Server. Create a new file ending with .properties in the gateway specification directory, and ensure it is only readable by the enveil user. A sample file, which you can use as a template, is installed to /etc/enveil/zeroreveal-client/gateway.sample.properties.

To begin, set the following properties:

enveil.client.gateway.rest.uri
The URI of the ZeroReveal Server to connect to; get this from the ZeroReveal Server administrator.

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

Verdict:

PASS.

5.4.1.12.5 FCS_TLSC_EXT.1.3 TSS

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

The relevant information is found in the following section(s): TOE Summary Specification FCS_TLSC_EXT.1

Upon investigation, the evaluator found that the TSS states that: The TOE performs X.509v3 certification validation. The TOE will reject trusted channel establishment if the certificate is invalid.

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

Verdict:

PASS.

5.4.1.12.1 FCS_TLSC_EXT.1.3 AGD

According to the PP, there are no AA requirements for AGD activity for this SFR.

5.4.1.13 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

5.4.1.13.1 FCS_TLSC_EXT.2.1 TSS

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

The relevant information is found in the following section(s): TOE Summary Specification FCS_TLSC_EXT.2.

Upon investigation, the evaluator found that the TSS states that: The TOE acts as a TLS client when establishing connection to LDAP directory for authentication and when establishing connection to ZeroReveal Compute Fabric Server for operation requests and responses.

When acting as a TLS client, the TOE supports mutual authentication using X.509v3 certificates. The TOE performs X.509v3 certification validation. The TOE will reject trusted channel establishment if the certificate is invalid.

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

Verdict:

PASS.

5.4.1.13.2 FCS_TLSC_EXT.2.1 AGD

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 required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.

Evaluator Findings:

- The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1"and ensured that it includes any instructions necessary to configure the TOE to perform mutual authentication.

- The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1"and ensured that the guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.

The relevant information is found in the following section(s):

- "3.4 Configuring ZeroReveal Client" of ZeroReveal Client Guide
- "4 Mutual TLS Configuration"
- "4.1 Certificate Requirements"
- "4.3 ZeroReveal Client LDAP Configuration"
- "6.2 LDAP Configuration"
- "6.3 Connecting ZeroReveal Client and ZeroReveal Server"

Upon investigation, the evaluator found that the AGD activity states that: For the configuration to be Common Criteria compliant, the key stores must be in the bcfks format. See Creating a bcfks from a p12 for instructions on converting key stores to the bcfks format.

Furthermore, certificates must contain the hostname or the IP address of the ZeroReveal Client's host machine as a Subject Alternative Name. See Mutual TLS Configuration for other certificate requirements. To generate a TLS keypair in a Common Criteria compliant fashion using ZeroReveal Client, consult Using ZeroReveal to Generate Certificate Signing Requests. The following are relevant settings in client.conf. The first two certificates mentioned above are configured in the client.conf file in a single key store using the properties below. The trust store needs to contains certificates that are trusted by the ZeroReveal Client. The trust store is used to validate the certificates of the ZeroReveal Server and the LDAP server. The last certificate above is used here: Connecting ZeroReveal Client and ZeroReveal Server.

enveil.security.tls.keystore.path

Path to the key store on ZeroReveal Server's local disk.

enveil.security.tls.keystore.type

Type of the key store (possible options are jks, pkcs12, or bcfks).

enveil.security.tls.keystore.password

The key store's password.

enveil.security.tls.truststore.path

Path to the trust store on ZeroReveal Server's local disk.

enveil.security.tls.keystore.type

Type of the key store (possible options are jks or bcfks).

enveil.security.tls.truststore.password

The trust store's password.

If the certificate keys are generated using Elliptic Curve Cryptography, ensure that the curve used is either secp256r1 or secp384r1. If RSA keys are used, they must be 2048, 3072 bits, or 4096 bits.

Ensure that all TLS key stores and TLS trust stores are stored in /etc/enveil/zeroreveal-client/certs/ and are readable only by the enveil user.

Make sure that the following Common Criteria specific options in client.conf are set with the following constraints:

enveil.client.auth.mechanisms

Comma-separated list of authentication mechanisms to use. Must be set to [certificate].

enveil.client.auth.require.user.cert

(boolean) Whether to require users to present valid TLS client certificates. Must be set to true

enveil.client.auth.certificate.user.source.mechanisms

(string) A comma-separated list of user stores for use with certificate authentication. Must be set to [ldap].

enveil.client.auth.certificate.ldap.ssl.enabled

(boolean) Whether to connect to the LDAP server under TLS for certificate enveil.client.auth. Must be set to true.

enveil.client.auth.certificate.ldap.connect.with.sasl.external

(boolean) Whether to authenticate to the LDAP server using a TLS client certificate or not for certificate auth ; see Certificate. Must be set to true

enveil.security.cert.revocation.check.mode
(string) Whether to check for certificate revocation using any provided CRL endpoint. Defaults to NONE.
Must be set to "HARD_FAIL".

ZeroReveal Client automatically restricts all TLS connections to TLS version 1.2, denying all other TLS versions. No further configuration is required to configure the cryptographic engine beyond the steps above in this guide. *Note:* The combination of the settings for enveil.client.auth.mechanisms and enveil.client.auth.require.user.cert ensures that ZeroReveal Client will only accept connections with mutual TLS.

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

Verdict:

PASS.

5.4.1.14   FCS_TLSC_EXT.3 TLS Client Support for Signature Algorithms Extension

5.4.1.14.1 FCS_TLSC_EXT.3.1 TSS

Objective:

The evaluator shall verify that TSS describes the signature_algorithm extension and whether the required behavior is performed by default or may be configured.

Evaluator Findings:

- The evaluator reviewed the TSS to ensure that it describes the signature_algorithm extension and the required behavior is performed by default or may be configured.
- The relevant information is found in the following section(s): TOE Summary Specification FCS_TLSC_EXT.3 and FCS_TLSC_EXT.1.

Upon investigation, the evaluator found that the TSS states that: The TOE presents the signature_algorithm extension in the client_Hello message with a supported_signature_algorithms value containing only the SHA-384 and SHA-512 hash algorithms.

The TOE supports SHA384 and SHA512 signature hash algorithms after having configured the TOE according to the [AGD].

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

Verdict:

PASS

5.4.1.14.2 FCS_TLSC_EXT.3.1 AGD

Objective:

If the TSS indicates that the signature_algorithm extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the signature_algorithm extension.

Evaluator Findings:

The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured that the guidance includes configuration of the signature_algorithm extension.

The relevant information is found in the following section(s):

- "3.4 Configuring ZeroReveal Client" of ZeroReveal Client Guide
- "4.2.1 Signature Algorithms as a TLS Client"

Upon investigation, the evaluator found that the AGD activity states that: Make sure that the following Common Criteria specific options in client.conf are set with the following constraints:

enveil.security.tls.niap.signature.algorithms

(boolean) Only used NIAP-approved signature algorithms. Must be set to true.

In order to follow Common Criteria, a requirement exists that demands any $ClientHello$ that originates from ZeroReveal Client or ZeroReveal Server presents a restricted set of $signature\_algorithms$ in the $supported\_signature\_algorithms$ extension. Algorithms using $SHA256$ are no longer allowed, which means that only algorithms using $SHA384$ or $SHA512$ may be used. These extensions are restricted by setting a system property ($jdk.tls.client.SignatureSchemes$). An example of an allowable value contained in this property is $ecdsa\_secp384r1\_sha384$ or $rsa\_pss\_rsae\_sha512$.

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

Verdict:

PASS.

5.4.1.15   FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension

5.4.1.15.1   FCS_TLSC_EXT.5.1 TSS

Objective:

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

Evaluator Findings:

- The evaluator reviewed the TSS to ensure that it describes the Supported Groups Extension
- The relevant information is found in the following section(s): TOE Summary Specification FCS_TLSC_EXT.5 and FCS_TLSC_EXT.1.

Upon investigation, the evaluator found that the TSS states that: The TOE implements the supported_Groups extension with groups secp384r1 and no others.

The TOE supports Elliptic Curves Extension in the Client Hello with the secp384r1 NIST curve. The supported curves are hardcoded and there are no configuration options.

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

Verdict:

PASS.

5.4.1.15.1   FCS_TLSC_EXT.5.1 AGD

According to the PP, there are no AA requirements AGD activity for this SFR.

5.4.1.16 FCS_TLSS_EXT.1 TLS Server Protocol

5.4.1.16.1 FCS_TLSS_EXT.1.1 TSS

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 reviewed the TSS 6 to ensure that the cipher suites supported are specified.

- The evaluator reviewed the TSS 6 to ensure that the cipher suites specified include those listed for this component.

The relevant information is found in the following section(s): TOE Summary Specification FCS_TLSS_EXT.1.

Upon investigation, the evaluator found that the TSS states that: When acting as a TLS server, the TOE implements TLSv1.2 and rejects all older versions of TLS and SSL, and supports the following cipher suites:

• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

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

Verdict:

PASS.

5.4.1.16.2 FCS_TLSS_EXT.1.1 AGD

Objective:

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

Evaluator Findings:

The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.

The relevant information is found in the following section(s):

- "3.4 Configuring ZeroReveal Client" of ZeroReveal Client Guide

Upon investigation, the evaluator found that the AGD activity states that: Make sure that the following Common Criteria specific options in client.conf are set with the following constraints:

enveil.security.tls.strict

(string) If true, requires TLSv1.2 and one of the following cipher suites for all connections: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384. (Note that the elliptic curve used with these cipher suites for key establishment is only $secp384r1$.) If false, accepts any valid TLS protocol and cipher suite available in the local Java installation. Must be set to true.

ZeroReveal Client automatically restricts all TLS connections to TLS version 1.2, denying all other TLS versions. No further configuration is required to configure the cryptographic engine beyond the steps in this guide. Note: The combination of the settings for enveil.client.auth.mechanisms and enveil.client.auth.require.user.cert ensures that ZeroReveal Client will only accept connections with mutual TLS.

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

Verdict:

PASS.

5.4.1.16.3 FCS_TLSS_EXT.1.2 TSS

Objective:

The evaluator shall verify that the TSS contains a description of the denial of old SSL and TLS versions consistent relative to selections in FCS_TLSS_EXT.1.2.

Evaluator Findings:

The evaluator reviewed the TSS to ensure that it contains a description of the denial of old SSL and TLS versions consistent relative to selections in FCS_TLSS_EXT.1.2.

The relevant information is found in the following section(s): TOE Summary Specification FCS_TLSS_EXT.1.

Upon investigation, the evaluator found that the TSS states that: When acting as a TLS server, the TOE implements TLSv1.2 and rejects all older versions of TLS and SSL, and supports the following cipher suites:

• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

• TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

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

Verdict:

PASS.

5.4.1.16.4 FCS_TLSS_EXT.1.2 AGD

Objective:

The evaluator shall verify that the AGD guidance includes any configuration necessary to meet this requirement.

Evaluator Findings:

The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured that it includes any configuration necessary to meet this requirement.

The relevant information is found in the following section(s):

   • "3.4 Configuring ZeroReveal Client" of ZeroReveal Client Guide

Upon investigation, the evaluator found that the AGD activity states that: Make sure that the following Common Criteria specific options in client.conf are set with the following constraints:

enveil.security.tls.strict

(string) If true, requires TLSv1.2 and one of the following cipher suites for all connections: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384. (Note that the elliptic curve used with these cipher suites for key establishment is only secp384r1.) If false, accepts any valid TLS protocol and cipher suite available in the local Java installation. Must be set to true.

ZeroReveal Client automatically restricts all TLS connections to TLS version 1.2, denying all other TLS versions. No further configuration is required to configure the cryptographic engine beyond the steps in this guide. Note: The combination of the settings for enveil.client.auth.mechanisms and enveil.client.auth.require.user.cert ensures that ZeroReveal Client will only accept connections with mutual TLS.

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

Verdict:

PASS.

5.4.1.16.5 FCS_TLSS_EXT.1.3 TSS

Objective:

The evaluator shall verify that the TSS describes the key agreement parameters of the server's Key Exchange message.

Evaluator Findings:

The evaluator reviewed the TSS to ensure that it describes the key agreement parameters of the server's Key Exchange message.

The relevant information is found in the following section(s): TOE Summary Specification FCS_TLSS_EXT.1.

Upon investigation, the evaluator found that the TSS states that: When acting as a TLS server, the TOE performs ECDH key establishment using the secp384r1 elliptic curve.

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

Verdict:

PASS.

5.4.1.16.6 FCS_TLSS_EXT.1.3 AGD

Objective:

The evaluator shall verify that any configuration guidance necessary to meet the requirement must be contained in the AGD guidance.

Evaluator Findings:

The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured that it includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS.

The relevant information is found in the following section(s):

- "3.4 Configuring ZeroReveal Client" of ZeroReveal Client Guide

Upon investigation, the evaluator found that the AGD activity states that: Make sure that the following Common Criteria specific options in client.conf are set with the following constraints:

enveil.security.tls.strict

(string) If true, requires TLSv1.2 and one of the following cipher suites for all connections: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384. (Note that the elliptic curve used with these cipher suites for key establishment is only secp384r1.) If false, accepts any valid TLS protocol and cipher suite available in the local Java installation. Must be set to true.

ZeroReveal Client automatically restricts all TLS connections to TLS version 1.2, denying all other TLS versions. No further configuration is required to configure the cryptographic engine beyond the steps in this guide.

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

Verdict:

PASS.

5.4.1.17 FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication

5.4.1.17.1 FCS_TLSS_EXT.2.1

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.4.1.17.2 FCS_TLSS_EXT.2.2 TSS (Applied TD0770)

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. If error messages are provided prior to terminating a session, the evaluator shall ensure the error messages are described.

Evaluator Findings:

- The evaluator reviewed the TSS to ensure that the description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication. If error messages are provided prior to terminating a session, the evaluator shall ensure the error messages are described.

The relevant information is found in the following section(s): TOE Summary Specification FCS_TLSS_EXT.2.

Upon investigation, the evaluator found that the TSS states that: The TOE acts as a TLS server when accepting HTTPS connection requests from an end user. When acting as a TLS server, the TOE supports mutual authentication using X.509v3 certificates. The TOE performs X.509v3 certification validation. The TOE will reject trusted channel establishment if the certificate is invalid.

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

Verdict:

PASS.

5.4.1.17.3 FCS_TLSS_EXT.2.2 AGD (Applied TD0770)

Objective:

- The evaluator shall verify that the AGD required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.

- The evaluator shall ensure that the AGD guidance includes instructions for configuring the server to require mutual authentication of clients using these certificates.

Evaluator Findings:

- The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured that the guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.

- The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" guidance to ensure it includes instructions for configuring the server to require mutual authentication of clients using these certificates

The relevant information is found in the following section(s):

- "3.4 Configuring ZeroReveal Client" of ZeroReveal Client Guide
- "4 Mutual TLS Configuration"
- "4.1 Certificate Requirements"
- "4.2.2 Signature Algorithms as a TLS Server"

Upon investigation, the evaluator found that the AGD activity states that: For the configuration to be Common Criteria compliant, the key stores must be in the bcfks format. See Creating a bcfks from a p12 for instructions on converting key stores to the bcfks format. See *Mutual TLS Configuration* for other certificate requirements. Furthermore, certificates must contain the hostname or the IP address of the ZeroReveal Client's host machine as a Subject Alternative Name. To generate a TLS keypair in a Common Criteria compliant fashion using ZeroReveal Client, consult Using ZeroReveal to Generate Certificate Signing Requests. The following are relevant settings in client.conf. The first two certificates mentioned above are configured in the client.conf file in a single key store using the properties below. The trust store needs to contains certificates that are trusted by the ZeroReveal Client.

enveil.security.tls.keystore.path

Path to the key store on ZeroReveal Server's local disk.

enveil.security.tls.keystore.type

Type of the key store (possible options are jks, pkcs12, or bcfks).

enveil.security.tls.keystore.password

The key store's password.

Ensure that all TLS key stores and TLS trust stores are stored in /etc/enveil/zeroreveal-client/certs/ and are readable only by the enveil user.

Make sure that the following Common Criteria specific options in client.conf are set with the following constraints:

enveil.common.niap.enforce

(boolean) Enforces that the server is configured to meet the NIAP requirements. Must be set to true.

enveil.client.auth.mechanisms

Comma-separated list of authentication mechanisms to use. Must be set to [certificate].

enveil.client.auth.require.user.cert

(boolean) Whether to require users to present valid TLS client certificates. Must be set to true.

enveil.security.tls.keystore.check

(boolean) Validates the key store on startup. Must be set to true.

enveil.security.tls.strict

(string) If true, requires TLSv1.2 and one of the following cipher suites for all connections: TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384. (Note that the elliptic curve used with these cipher suites for key establishment is only secp384r1.) If false, accepts any valid TLS protocol and cipher suite available in the local Java installation. Must be set to true.

enveil.security.tls.client.certificate.check

(boolean) Whether to check the validity of a certificate presented by any TLS client (currently only ZeroReveal Client). Must be set to true.

enveil.security.tls.niap.signature.algorithms

(boolean) Only used NIAP-approved signature algorithms. Must be set to true.

enveil.security.cert.revocation.check.mode

(string) Whether to check for certificate revocation using any provided CRL endpoint. Defaults to

NONE.

Must be set to "HARD_FAIL".

ZeroReveal Client automatically restricts all TLS connections to TLS version 1.2, denying all other TLS versions. No further configuration is required to configure the cryptographic engine beyond the steps above in this guide. Note: The combination of the settings for enveil.client.auth.mechanisms and enveil.client.auth.require.user.cert ensures that ZeroReveal Client will only accept connections with mutual TLS.

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

Verdict:

PASS.

5.4.1.17.4 FCS_TLSS_EXT.2.3 TSS

Objective:

If the product implements mutual authentication, the evaluator shall verify that the TSS describes how the DN and SAN in the certificate is compared to the expected identifier.

Evaluator Findings:

The evaluator reviewed the TSS to ensure that it describes how the DN and SAN in the certificate is compared to the expected identifier.

The relevant information is found in the following section(s): TOE Summary Specification FCS_TLSS_EXT.2.

Upon investigation, the evaluator found that the TSS states that: When acting as a TLS server, the TOE supports mutual authentication using X.509v3 certificates. The TOE validates the presented reference identifier in accordance with RFC 6125, and permits the reference identifier to be the CN, DN, or SAN-DNS. Where present, the SAN-DNS identifier supersedes the DN or CN values. When acting as a server, the TOE does not accept wildcards.

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

Verdict:

PASS.

5.4.1.17.5 FCS_TLSS_EXT.2.3 AGD

Objective:

If the DN is not compared automatically to the domain name, IP address, username, or email address, the evaluator shall ensure that the AGD guidance includes configuration of the expected identifier or the directory server for the connection.

Evaluator Findings:

The evaluator checked the AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" "and ensured that it guidance includes configuration of the expected identifier or the directory server for the connection.

The relevant information is found in the following section(s):

- "3.4 Configuring ZeroReveal Client "of ZeroReveal Client Guide
- "5.3 Usage"

Upon investigation, the evaluator found that the AGD activity states that: For the configuration to be Common Criteria compliant, the key stores must be in the bcfks format. See *Creating a bcfks from a p12* for instructions on converting key stores to the bcfks format. Furthermore, certificates must contain the hostname or the IP address of the ZeroReveal Client's host machine as a Subject Alternative Name. When creating your certificate, make sure it contains the hostname or IP address of the ZeroReveal Client host machine as a Subject Alternative Name (SAN).

Both ZeroReveal Server and ZeroReveal Client include a script called csr-utility which uses the ZeroReveal DRBG and cryptographic libraries to generate RSA and Elliptic Curve keypairs then prepare a basic Certificate Signing Request for signing into a certificate by a Certificate Authority.

Submit the CSR to the appropriate Certificate Authority for signing. By default it will have a nondescript and invalid Distinguished Name. As a result, the Certificate Authority will need to ensure that issued certificates contain Subject Alternative Names of the hostname or the IP address of the ZeroReveal Client or ZeroReveal Server host machine the certificate is intended for.

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

Verdict:

PASS.

5.4.1.18   TLS Server Support for Signature Algorithms Extension

5.4.1.18.1 FCS_TLSS_EXT.3.1 TSS

Objective:

The evaluator shall verify that TSS describes the supported_signature_algorithms field of the Certificate Request and whether the required behavior is performed by default or may be configured.

Evaluator Findings:

- The evaluator reviewed the TSS to ensure that it describes the signature_algorithm extension and the required behavior is performed by default or may be configured.

- The relevant information is found in the following section(s): TOE Summary Specification FCS_TLSS_EXT.3.

Upon investigation, the evaluator found that the TSS states that: The TOE supports SHA256 and SHA384 signature hash algorithms.

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

Verdict:

PASS

5.4.1.18.2 FCS_TLSS_EXT.3.1 AGD

Objective:

If the TSS indicates that the signature_algorithm field must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the signature_algorithm field.

Evaluator Findings:

The evaluator checked the AGD **"**ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" and ensured that the guidance includes configuration of the signature_algorithms field

- 3.4 Configuring ZeroReveal Client
- 4.2.1 Signature Algorithms as a TLS Server

Upon investigation, the evaluator found that the AGD activity states that: Make sure that the following Common Criteria specific options in client.conf are set with the following constraints:

- o   enveil.security.tls.niap.signature.algorithms

(boolean) Only used NIAP-approved signature algorithms, Must be set to true.

In order to follow Common Criteria, a requirement exists that demands any CertificateRequest that originates from ZeroReveal (when acting as a TLS server) presents a restricted set of signature_algorithms in the supported_signature_algorithms extension. Only algorithms using SHA256 or SHA384 may be used. These extensions are restricted by setting a system property (jdk.tls.server.SignatureSchemes).

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

Verdict:

PASS

5.4.2 Identification and Authentication (FIA)

5.4.2.1   FIA_X509_EXT.1 X.509 Certificate Validation

5.4.2.1.1   FIA_X509_EXT.1.1 TSS

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

Evaluator Findings:
- The evaluator reviewed the TSS 6 and ensured that it describes where the check of validity of the certificates takes place.
- The evaluator reviewed the TSS 6 and ensured that it provides a description of the certificate path validation algorithm.

The relevant information is found in the following section(s): TOE Summary Specification FIA_X509_EXT.1.

Upon investigation, the evaluator found that the TSS states that: The TOE uses X.509v3 certificates to authenticate network endpoints for the HTTPS/TLS trusted channel communications. The TOE complies with RFC 5280 by implementing all SHALL, SHOULD, and MUST statements and not implementing any SHALL NOT, SHOULD NOT, or MUST NOT statements.

The TOE uses the Java PKIX and Bouncy Castle validation tools. The notBefore and notAfter dates included in certificates will be checked to be before and after the current time respectively. Certificates received as part of TLS connections are checked for a valid path up to the certificate authority roots (which must have the X509v3 Basic Constraint CA: True). The TOE performs all of the required checks on trust path requirements, CA validity, key usages, and extended key usages. In the process, it ensures certificates presented for client authentication have the digitalSignature keyUsage and TLS Client extendedKeyUsage.

CRL checking as specified in RFC 8603 revocation checking will be attempted on certificates that have listed distribution points. It is a configuration option for administrators to decide if failure to determine a certificate's status (if that certificate lists an endpoint and the endpoint is unreachable) should result in certificate rejection. Enveil enables this platform-provided functionality by adding the java.security.cert.PKIXRevocationChecker class to the chain of X509 TrustManagers associated with TLS contexts used to form connections.

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

Verdict:
PASS.

5.4.2.1.1   FIA_X509_EXT.1.1 AGD

According to the PP, there are no AA requirements for this AGD activity for this SFR.

5.4.2.1.2   FIA_X509_EXT.1.2

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.4.2.2   FIA_X509_EXT.2 X.509 Certificate Authentication

5.4.2.2.1   FIA_X509_EXT.2.1 TSS

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.

- 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, the evaluator shall ensure that the AGD contains instructions on how this configuration action is performed.

Evaluator Findings:

- The evaluator reviewed the TSS 6 and ensured that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates.

- The evaluator reviewed the TSS 6 and ensured 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 reviewed the TSS 6 and ensured that any distinctions between trusted channels are described.

- The evaluator reviewed the TSS 6 and ensured that it, if the requirement that the administrator is able to specify the default action, the AGD contains instructions on how this configuration action is performed.

The relevant information is found in the following section(s): TOE Summary Specification FIA_X509_EXT.2

Upon investigation, the evaluator found that the TSS states that: The TOE uses X.509v3 certificates for TLS mutual authentication with REST API clients, GUI clients, and connections to the ZR Server and to LDAP servers. An administrator sets the certificate to be used for each distinct purpose in the TOE configuration file. When presented with an invalid certificate, the connections are rejected.

The evaluator examined the section titled "3.4 Configuring ZeroReveal Client*"* and "4.1 Certificate Requirements" **"**ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" to verify that it describes configuring the operating environment so that the TOE can use the certificates. Upon investigation, the evaluator found that the AGD states that:

Ensure that all TLS key stores and TLS trust stores are stored in /etc/enveil/client/certs/ and are readable only by the enveil user.

Make sure that the following Common Criteria specific options in client.conf are set with the following constraints:

enveil.common.niap.enforce

(boolean) Enforces that the server is configured to meet the NIAP requirements. Must be set to true.

enveil.client.auth.mechanisms

Comma-separated list of authentication mechanisms to use. Must be set to [certificate].

enveil.client.auth.require.user.cert

(boolean) Whether to require users to present valid TLS client certificates. Must be set to true.

enveil.security.tls.keystore.check

(boolean) Validates the key store on startup. Must be set to true.

enveil.security.tls.client.certificate.check

(boolean) Whether to check the validity of a certificate presented by any TLS client (currently only ZeroReveal Client). Must be set to true.

enveil.security.cert.revocation.check.mode

(string) Whether to check for certificate revocation using any provided CRL endpoint. Defaults to

NONE.

Must be set to "HARD_FAIL".

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

Verdict:

PASS.

5.4.2.2.1   FIA_X509_EXT.2.1 AGD

According to the PP, there are no AA requirements for this AGD activity for this SFR.

5.4.3   Protection of the TSF (FPT)

5.4.3.1   FPT_TUD_EXT.2 Integrity for Installation and Update

5.4.3.1.1   FPT_TUD_EXT.2.1

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.4.3.1.2   FPT_TUD_EXT.2.2

According to the PP, there are no AA requirements for TSS and AGD activities for this SFR.

5.4.3.1.3   FPT_TUD_EXT.2.3 TSS

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 reviewed the TSS 6 and ensured that it identifies how the application installation package is signed by an authorized source. The definition of an authorized source must be contained in the TSS.

The relevant information is found in the following section(s): TOE Summary Specification FPT_TUD_EXT.2

Upon investigation, the evaluator found that the TSS states that: The TOE supports packages running on Red Hat and Red Hat derivatives in RPM format. Official Enveil RPMs are signed using Enveil's private signing key. When using Yum to install Enveil TOE packages, the GPG signatures on the RPM files will automatically be checked. If they are missing a signature or signed with the wrong GPG key, then an error indicating that the GPG keys for the repository do not match the package will be displayed and the installation will automatically abort. These checks are also run during the installation of every update.

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

Verdict:

PASS.

5.4.3.1.4   FPT_TUD_EXT.2.3 AGD

According to the PP, there are no AA requirements for this AGD activity for this SFR.

6 Security Assurance Requirements

6.1 Security Target (ASE)

There are no new Assurance Activities included in the PP for ASE.

6.2 Development (ADV)

6.2.1 ADV_FSP.1 Basic Functional Specification

6.2.1.1 ADV_FSP.1.1E

Objective:

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Evaluator Findings:

The evaluator confirmed that the information provided meets all requirements for content and presentation of evidence.

Verdict:

PASS.

6.2.1.2 ADV_FSP.1.2E

Objective:

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

Evaluator Findings:

The evaluator determined that the functional specification is an accurate and complete instantiation of the SFRs.

Verdict:

PASS.

6.3 Guidance Documentation (AGD)

6.3.1 AGD_OPE.1 Operational User AGD

6.3.1.1 AGD_OPE.1.1E

Objective:

- The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
  - Some of the contents of the AGD will be verified by the evaluation activities in Section 5.1 Security Functional Requirements and evaluation of the TOE according to the [CEM]. The following additional information is also required.
  - If cryptographic functions are provided by the TOE, the AGD shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

- The documentation must describe the process for verifying updates to the TOE by verifying a digital signature – this may be done by the TOE or the underlying platform.
- The evaluator shall verify that this process includes the following steps:
  - Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).
  - Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the digital signature. The TOE will likely contain security functionality that does not fall in the scope of evaluation under this PP. The AGD shall make it clear to an administrator which security functionality is covered by the evaluation activities.

Evaluator Findings:

- The evaluator examined the section titled "**3.2 Prerequisites**" of **"ZeroReveal Client Guide"** AGD **"**ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" in the AGD to verify that it contains instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE.

- The evaluator examined the section titled "**3.3 Installing ZeroReveal Client**" in the AGD to verify that it describes the process for verifying updates to the TOE by verifying a digital signature.

- The evaluator examined the section titled "**3.5 Determining the Installed Version of ZeroReveal Client**" and "**3.6 Updating ZeroReveal Client**" in the AGD and found that AGD describes the instructions for obtaining the updates.

- Section "**Security Target Introduction**" of the ST was used to determine if there is any functionality excluded from the TOE. Section "**Product Functionality not Included in the Scope of the Evaluation**" in the ST states that the homomorphic encryption process, including the algorithms, uses and the security strength of the resultant ciphertext and access to the local configuration files is excluded in from the evaluation and only the default is supported. The evaluator then examined the**" 1.1 Targets of Evaluation and Scope"** of **"ZeroReveal Client Guide"** AGD **"**ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1 and determined it advises the user that the ZeroReveal Client and ZeroReveal Server are evaluated as software applications only. For emphasis, the following are outside of the scope of this evaluation:

• The homomorphic encryption techniques used for the ZeroReveal Client and ZeroReveal Server operations.
• The interface used to modify the ZeroReveal Client and ZeroReveal Server configuration files.

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

Verdict:

PASS.

6.3.2 Preparative Procedures (AGD_PRE.1)

6.3.2.1   AGD_PRE.1.1E

Objective:

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Evaluator Findings:

The evaluator confirmed the section "**3.2 Prerequisites**" of "**ZeroReveal Client Guide**" AGD "ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1" that the information provided meets all requirements for content and presentation of evidence.

Verdict:

PASS.

6.3.2.2   AGD_PRE.1.2E

Objective:

The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

- The evaluator shall confirm that the documentation describes how to configure the TOE platform.

- The evaluator shall confirm that the documentation describes how to configure the TOE's Operational Environment.

Evaluator Findings:

- The evaluator examined section "**3.1 System Requirements**" of "**ZeroReveal Client Guide**" AGD *"ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1"* to identify the supported platform of the TOE.

- The evaluator examined section "**1.3.1 Physical Boundary**" of the ST and identified the operational environment.

The evaluator examined section "**3.2 Prerequisites**" of "**ZeroReveal Client Guide**" AGD and determined configuration requirements of the TOE platform required having Rocky version 8.7 with SELinux and Amazon Corretto Java 8 Runtime Environment patch version >= 201 installed.

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

Verdict:

PASS.

6.4   Life-cycle Support (ALC)

6.4.1 ALC_CMC.1 Labeling of the TOE

6.4.1.1   ALC_CMC.1.1E

Objective:

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

- 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.

- The evaluator shall check the AGD guidance to ensure that the version number is consistent with that in the ST.

- The evaluator shall check the TOE samples received for testing to ensure that the version number is consistent with that in the ST.

- If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

Evaluator Findings:

- The evaluator examined the ST section titled "Security Target and TOE Identifier, TOE Reference, TOE Identification" to verify that the ST contains an identifier "Enveil ZeroReveal® Compute Fabric Client v4.6.3" that specifically identifies the version that meets the requirements of the ST.

- The evaluator examined the AGD, the Cover page of ZeroReveal® Compute Fabric Configuration Guide for Common Criteria v3.1 to verify that the AGD contains an identifier "Enveil ZeroReveal® Compute Fabric Client v4.6.3" that specifically identifies the version number is consistent with that in the ST.

- The evaluator examined the vendor's website "www.enveil.com" and determined the vendor does not maintain a website advertising a Common Criteria evaluated version of their product.

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

Verdict:

PASS.

6.4.2 ALC_CMS.1 TOE CM Coverage

6.4.2.1   ALC_CMS.1.1E

Objective:

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

- The evaluator shall ensure that the developer has identified (in AGD for application developers concerning the targeted platform) one or more development environments appropriate for use in developing applications for the developer's platform. For each of these development environments, the developer shall provide information on how to configure the environment to ensure that buffer overflow protection mechanisms in the environment(s) are invoked (e.g., compiler flags).

- The evaluator shall ensure that this documentation also includes an indication of whether such protections are on by default, or have to be specifically enabled.

- The evaluator shall ensure that the TSF is uniquely identified (with respect to other products from the TSF vendor), and that documentation provided by the developer in association with the requirements in the ST is associated with the TSF using this unique identification.

Evaluator Findings:

- The evaluator examined FPT_AEX_EXT.1 in the TSS section of the ST and determined that the Enveil's application and libraries included in the TOE are compiled with flags that protect buffer overflow. The flags are automatically enabled by invoking a script, used to compile the Java application.

- The evaluator examined the documentation received by the vendor and determined that the TOE is Enveil's only product.

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

Verdict:

PASS.

6.5   Tests (ATE)

6.5.1   ATE_IND.1 Independent Testing

6.5.1.1   ATE_IND.1.1E

Objective:

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Evaluator Findings:

The evaluator confirmed that the information provided meets all requirements for content and presentation of evidence.

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

Verdict:

PASS.

6.5.1.2   ATE_IND.1.2E

Objective:

The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

- 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:

An evaluator prepared a test plan that included for each SFR included in the ST, the tests required for that SFR as found in the PP and Functional Package. Specifically, the Test Plan included for each test:

- o The identity of the SFR and unique test number,

- o Test Assurance Activity – the test definition found in the PP or Functional Package,

- o Test Steps - the test steps required to complete the test,

- o Expected Test Results - a description of the expected test results,

- o Test Output – a place holder for screenshots and wireshark images from running the specific test.

- o Pass/Fail with Explanation – a place holder to report whether the test passed or failed and an explanation of the result.

- The evaluator then examined the "**Physical Boundary**" section in the ST and the following sections to determine what the TOE is and what is required in the TOE's operational environment. The evaluator determined based on the SFRs and the "TOE Operational Environment" figure in the [ST], the connections that needed to be configured and tested and the tools that are required. The evaluator then created a detailed diagram of the "TOE Operational Environment" figure in the [ST] identifying the TOE, the operational environment, and the required test equipment. Each system in the diagram, the Test Bed, included an IP address.

- Additionally, the evaluator created a detailed table identifying the TOE, the OE, and the hardware/software required for testing.

- The evaluator then set up the Test Bed. The AGD was used to install and configure the TOE.

- Another evaluator then examined the completed Test Report that included for each SFR included in the ST, the tests required for that SFR as found in the PP and Functional Package. Specifically, the Test Plan included for each test:

- o A diagram and description of the Test Bed including all required test equipment and IP addresses.

- o The identity of the SFRs and unique test number,

- o Test Assurance Activity – the test definition found in the PP or Functional Package,

- o Test Steps - the test steps required to complete the test,

- o Expected Test Results - a description of the expected test results,

- o Test Output – the screenshots and wireshark images from running the specific test.

- o Pass/Fail with Explanation – a report on whether the test passed or failed and an explanation of the result.

- All tests were reported as passed and there were no reported system crashes.

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

Verdict:

PASS.

6.6   Vulnerability Assessment (AVA)

6.6.1 AVA_VAN.1 Vulnerability Survey

Objective:

AVA_VAN.1.1E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.1.2E - The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.1.3E - The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

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

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

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

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

Evaluator Findings:

- The evaluator performed a *Vulnerability Assessment for Enveil ZeroReveal® Compute Fabric Client 4.6.3* on *April 04, 2024,* and generated a vulnerability report to document their findings with respect to this requirement.

- The evaluator examined the AVA document and determined that the report included a list of public search sites and search strings. The search strings included the product; the product vendor; the application's name; the libraries packaged with the TOE, specifically the crypto library included in the TOE; the API's invoked; the operating system; and the hardware platform. The evaluator concluded that the lists were reasonable for an application running on a Linux device. The evaluator concluded that the lists satisfied this work unit.

- The evaluator examined the AVA document and concluded that for each vulnerability found, the report included a determination if the vulnerability applied to the TOE and if it did, the action that occurred to remediate the vulnerability. If a vulnerability did not apply to the TOE, a rational of why the vulnerability did not apply to the TOE was included. The evaluator concluded that this work unit is satisfied.

- For Windows, Linux, macOS and Solaris: The evaluator performed the virus scans using ClamAV antivirus software on the Linux platform with the most current virus definitions against the application files and verified that no files are flagged as malicious. The scan was performed on 04/08/2024.

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

Verdict:

PASS.

7 Detailed Test Cases (Test Activities)

7.1 APP_1.4

7.1.1 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, • $|p-q| > 2^{nlen/2 - 100}$, • $p \geq 2^{nlen/2 - 1/2}$, • $q \geq 2^{nlen/2 - 1/2}$, • $2^{(nlen/2)} < d < LCM(p-1,q-1)$,

- e·d = 1 mod 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 #A4651 for FIPS186-4 RSA key generation using the key size 2048,3072 and 4096. This certificate provides assurance that the TSF performs these functions as required. **Key Generation for Elliptic Curve Cryptography (ECC)** The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for FIPS186-4 ECDSA key generation and key verification using the key sizeS P256 and P384. This certificate provides assurance that the TSF performs these functions as required. **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 The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for FIPS186-4 RSA key generation using the key size 2048, 3072 and 4096. This certificate provides assurance that the TSF performs these functions as required. **Key Generation for Elliptic Curve Cryptography (ECC)** Pass. The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for FIPS186-4 ECDSA key generation and key verification using the key sizeS P256 and P384. This certificate provides assurance that the TSF performs these functions as required. **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.

7.1.2 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** The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for Elliptic curve based key establishment (NIST SP 800-56A). This certificate provides assurance that the TSF performs these functions as required. **SP800-56B Key Establishment Schemes** SP800-56 tests are not applicable as the TOE does not use or claim SP800-56B key establishment schemes. **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** Pass The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for Elliptic curve based key establishment (NIST SP 800-56A). This certificate provides assurance that the TSF performs these functions as required. **SP800-56B Key Establishment Schemes** N/A because the TOE does not use or claim SP800-56B key establishment schemes. **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.

7.1.3 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 #A4651 for SHA2-256 (FIPS Pub 180-4), SHA2-384 (FIPS Pub 180-4) and SHA2-512 (FIPS Pub 180-4). This certificate provides assurance that the TSF performs these functions as required.
Pass/Fail with Explanation	Pass. The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for SHA2-256 (FIPS Pub 180-4), SHA2-384 (FIPS Pub 180-4) and SHA2-512 (FIPS Pub 180-4). This certificate provides assurance that the TSF performs these functions as required.

7.1.4 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 #A4651 for HMAC-SHA2-256 (FIPS Pub 198-1), HMAC-SHA2-384 (FIPS Pub 198-1) and HMAC-SHA2-512 (FIPS Pub 198-1). This certificate provides assurance that the TSF performs these functions as required.
Pass/Fail with Explanation	Pass. The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for HMAC-SHA2-256 (FIPS Pub 198-1), HMAC-SHA2-384 (FIPS Pub 198-1) and HMAC-SHA2-512 (FIPS Pub 198-1). This certificate provides assurance that the TSF performs these functions as required.

7.1.5 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** The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for FIPS186-4 ECDSA signature generation and signature verification (FIPS Pub 186-4) using the key sizes P256 and P384. This certificate provides assurance that the TSF performs these functions as required. **RSA Signature Algorithm Tests** The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for RSA signature generation and signature verification (FIPS Pub 186-4) using 2048, 3072 and 4096 bits RSA keys. This certificate provides assurance that the TSF performs these functions as required.
Pass/Fail with Explanation	**ECDSA Algorithm Tests** Pass. The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for FIPS186-4 ECDSA signature generation and signature verification (FIPS Pub 186-4) using the key sizes P256 and P384. This certificate provides assurance that the TSF performs these functions as required. **RSA Signature Algorithm Tests** Pass. The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for RSA signature generation and signature verification (FIPS Pub 186-4) using 2048, 3072 and 4096 bits RSA keys. This certificate provides assurance that the TSF performs these functions as required.

7.1.6 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:

$$CT[i] = AES\text{-}CBC\text{-}Encrypt(Key, PT)$$

$$PT = CT[i\text{-}1]$$

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**
	This is not applicable as the TOE does not claim or use AES in CBC mode.
	AES-CBC Multi-Block Message Test
	This is not applicable as the TOE does not claim or use AES in CBC mode.
	AES-GCM Monte Carlo Tests
	The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for AES-GCM (NIST SP800-38D) using key size 256 for encryption and decryption. This certificate provides assurance that the TSF performs these functions as required.
	AES-XTS Tests
	This is not applicable as the TOE does not claim or use AES in XTS mode.
	AES-CCM Tests
	The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for AES-CCM (NIST SP800-38C) using key size 256 for encryption and decryption. This certificate provides assurance that the TSF performs these functions as required.
	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**
	N/A because the TOE does not claim AES in CBC mode.
	AES-CBC Multi-Block Message Test
	N/A because the TOE does not claim AES in CBC mode.
	AES-GCM Monte Carlo Tests
	Pass. The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for AES-GCM (NIST SP800-38D) using key size 256 for encryption and decryption. This certificate provides assurance that the TSF performs these functions as required.
	AES-XTS Tests
	N/A because the TOE does not claim AES in XTS mode.
	AES-CCM Tests
	Pass. The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for AES-CCM (NIST SP800-38C) using key size 256 for encryption and decryption. This certificate provides assurance that the TSF performs these functions as required.
	AES-CTR Tests
	N/A because the TOE does not claim AES in CTR mode.

7.1.7 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 #A4651 for HMAC 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. The evaluator examined the ST and found that in Section "Cryptographic Support" that the TOE was awarded the CAVP certificate #A4651 for HMAC DRBG (NISP SP 800-90A) (AES-256). This certificate provides assurance that the TSF performs these functions as required.

7.1.8 FCS_RBG_EXT.2.2 Test #1

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

7.1.9 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	TOE as client to ZR Server • Configure the TOE to the TLS server (ZR Server) • Establish a connection with the TOE over TLS using the cipher ECDHE-RSA-AES256-GCM-SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture TOE as client to LDAP server • Configure the TOE to the TLS server (LDAP Server) • Establish a connection with the TOE over TLS using the cipher ECDHE-RSA-AES256-GCM-SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture

Item	Data
Expected Test Results	Establish a successful TLS connection with server and observed the traffic was traffic is encrypted.
Pass/Fail with Explanation	Pass. The TOE accepts the connection to the server with HTTPS and the traffic is encrypted. This meets the testing requirement

7.1.10 FCS_HTTPS_EXT.1.2/Client Test #1

Item	Data
Test Assurance Activity	Other tests are performed in conjunction with the TLS package.
Pass/Fail with Explanation	Pass. This testing was performed in conjunction with the TLS package. This meets the testing requirement.

7.1.11 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	- TOE as client to ZR Server • Valid certificate chain • Configure the TOE to the TLS server (ZR Server) • Create a full chain of certificates to connect to the TOE. • Upload a complete certificate validation chain to the TOE. • Attempt to connect to the TOE with the full chain of proper certificates • Verify the connection is successful • Verify the connection is successful via packet capture

	• Invalid certificate chain
	• Delete the ICA2 certificate from the chain
	• Attempt to connect to the TOE without the intermediate CA and verify the connection failed
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
	- TOE as client to LDAP server
	• Valid certificate chain
	• Configure the TOE to the TLS server (LDAP Server)
	• Create a full chain of certificates to connect to the TOE.
	• Upload a complete certificate validation chain to the TOE.
	• Attempt to connect to the TOE with the full chain of proper certificates
	• Verify the connection is successful
	• Verify the connection is successful via packet capture
	• Invalid certificate chain
	• Delete the ICA2 certificate from the chain
	• Attempt to connect to the TOE without the intermediate CA and verify the connection failed
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	• When a complete cert chain is present, a TLS connection can be established • When an incomplete cert chain is present, a TLS connection cannot be established
Pass/Fail with Explanation	Pass. The TOE accepts the connection when the entire certificate chain is available and rejects the connection when the entire certificate chain is not presented. This meets the test requirements.

7.1.12 FCS_HTTPS_EXT.1.1/Server Test #1

Item	Data

Test Assurance Activity	The evaluator shall attempt to establish an HTTPS connection to the TOE using a client, 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	TOE as GUI server • Log into the TOE via GUI • Verify that user is successfully logged in to the TOE. • Verify the connection succeeds, and traffic is encrypted with TLS TOE as HTTPS server via REST-API • Configure the TOE as HTTPS server via REST-API • Establish a connection with the TOE over TLS • Verify the connection was successful • Verify the connection succeeds, and traffic is encrypted with TLS
Expected Test Results	Establish a successful HTTPS connection to the TOE using a client and the traffic is encrypted.
Pass/Fail with Explanation	Pass. The TOE allows a client to successfully connect with HTTPS and the traffic is encrypted.

7.1.13 FCS_HTTPS_EXT.1.2/Server Test #1

Item	Data
Test Assurance Activity	Other tests are performed in conjunction with the TLS package.
Pass/Fail with Explanation	Pass. This testing was performed in conjunction with the TLS package. This meets the testing requirement.

7.1.14 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>TOE as GUI server</u> • Valid certificate chain • Create a full chain of certificates to connect to the TOE. • Upload a complete certificate validation chain to the TOE. • Attempt to connect to the TOE with the full chain of proper certificates • Verify the connection is successful • Verify the connection is successful via packet capture • Invalid certificate chain • Delete the root-ca certificate from the chain • Attempt to connect to the TOE without the CA and verify the connection failed • Verify the connection failure via packet capture • Verify the connection failure via logs <u>TOE as HTTPS server via REST-API</u> • Valid certificate chain • Create a full chain of certificates to connect to the TOE. • Upload a complete certificate validation chain to the TOE. • Attempt to connect to the TOE with the full chain of proper certificates • Verify the connection is successful • Verify the connection is successful via packet capture • Invalid certificate chain • Delete the ICA2 certificate from the chain • Attempt to connect to the TOE without the intermediate CA and verify the connection failed

	• Verify the connection failure via packet capture
	• Verify the connection failure via logs

Expected Test Results	• When a complete cert chain is present, a TLS connection can be established
	• When an incomplete cert chain is present, a TLS connection cannot be established
Pass/Fail with Explanation	Pass. The TOE accepts the connection when the entire certificate chain is presented and rejects the connection when the entire certificate chain is not presented. This meets the test requirements.

7.1.15 FCS_RBG_EXT.1.1 Test #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:Linux...
	The evaluator shall verify that the application collects random from /dev/random or /dev/urandom.
Pass/Fail with Explanation	NA. The ST does not select '**invokes platform-provided DRBG functionality**'.

7.1.16 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	• Find the path where the certificates and private keys (credentials) for TLS connection are stored
	• Verify that the credentials stored are encrypted using AES-CCM.

Item	Data
Expected Test Results	The application encrypts the stored credentials
Pass/Fail with Explanation	Pass. It is verified that the stored credentials are encrypted

7.1.17 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:Linux...** The evaluator shall verify that all keys are stored using Linux keyrings.
Pass/Fail with Explanation	NA. The ST does not select '**invokes platform-provided functionality'**.

7.1.18 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**
Test Steps	• Start the application and confirm status. • Note the locations where application writes data. • Note the path where the certificates and private keys (credentials) for TLS connection are stored. • Run the lsof command to determine if additional locations are written to (the command was run a few times to ensure that the lsof command could capture writes in conjunction with executing a connection between zeroreveal server and zeroreveal client). • Verify that locations where data was stored are encrypted.
Expected Test Results	The TOE verifies locations where application writes data are encrypted

Pass/Fail with Explanation	Pass. Locations where application writes data are encrypted. This meets testing requirements.

7.1.19 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:Linux...** The Linux platform currently does not provide data-at-rest encryption services which depend upon invocation by application developers. The evaluator shall verify that the Operational User Guidance makes the need to activate platform encryption clear to the end user.
Test Steps	Verify the Operational User Guidance mentions the need to activate platform encryption.
Expected Test Results	The Operational User Guidance mentions the need to activate platform encryption
Pass/Fail with Explanation	Pass. The evaluator confirmed that the Operational User Guidance makes the need to activate platform encryption clear to the end user.

7.1.20 FDP_DEC_EXT.1.1 Test #1

Item	Data
Test Assurance Activity	**Platforms:Linux...** The evaluator shall verify that either the application software or its documentation provides a list of the hardware resources it accesses.
Test Steps	• List the resources accessed by the application according to ST. • Verify the same with the documentation provided- ZeroReveal Compute Fabric Configuration Guide for Common Criteria v3.1- Section 3.1.
Expected Test Results	The application software or its documentation provides a list of the hardware resources it accesses.
Pass/Fail with Explanation	Pass. The resources accessed mentioned in the ST were verified with those in the documentation provided.

7.1.21 FDP_DEC_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	**Platforms:Linux...** The evaluator shall verify that either the application software or its documentation provides a list of sensitive information repositories it accesses.
Pass/Fail with Explanation	NA. The TOE does not access sensitive information repositories.

7.1.22 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	TOE as GUI server • Log into the TOE via GUI • Verify that user is successfully logged in to the TOE. • Verify the connection succeeds, and traffic is encrypted with TLS TOE as HTTPS server via REST-API • Configure the TOE as HTTPS server via REST-API • Establish a connection with the TOE over TLS • Verify the connection was successful • Verify the connection succeeds, and traffic is encrypted with TLS TOE as client to ZR Server • Configure the TOE to the TLS server • Establish a connection with the TOE over TLS using the cipher ECDHE-RSA-AES256-GCM-SHA384 • Verify the connection was successful

	• Verify the connection was successful via packet capture TOE as client to LDAP server • Configure the TOE to the TLS server • Establish a connection with the TOE over TLS using the cipher ECDHE-RSA-AES256-GCM-SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture
Expected Test Results	All the traffic captured when the TOE is exercised should be encrypted by either TLS or HTTPS
Pass/Fail with Explanation	Pass. Any network communications witnessed are documented in the TSS or are user initiated.

7.1.23 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	• Start the TOE application services
	
	• Use the nmap utility for port scanning to check the open ports for the below protocols
	
	    o TCP IPv4
	
	    o UDP IPv4
	
	• Compare the open ports from the above output with those specified in ST and verify they match.

Expected Test Results	Application should not open any unexpected ports when running.
Pass/Fail with Explanation	Pass. The TOE does not open any unexpected ports. This meets the testing requirements.

7.1.24 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	NA. The TOE is not installed with default credentials.

7.1.25 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	NA. The TOE is not installed with default credentials.

7.1.26 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	NA. The TOE is not installed with default credentials.

7.1.27 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:Linux... The evaluator shall run the command find -L . -perm /002 inside the application's data directories to ensure that all files are not world-writable. The command should not print any files.
Test Steps	• Start the TOE application service. • Find the path for data directories from client config file. • Ensure application data directories are not world-writable.
Expected Test Results	The TOE shall verify that all files are not world-writable and the command shall not print any files
Pass/Fail with Explanation	Pass. Application data directories are verified to be not world writable. This meets the testing requirements.

7.1.28 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:Linux...** The evaluator shall run the application while monitoring it with the utility strace. The evaluator shall make security-related changes to its configuration. The evaluator shall verify that strace logs corresponding changes to configuration files that reside in /etc (for system-specific configuration), in the user's home directory (for user-specific configuration), or /var/lib/ (for configurations controlled by UI and not intended to be directly modified by an administrator).
Test Steps	• Verify that TOE config files are not dynamically written. • Verify the status of the application. • Run the strace utility such that it monitors the TOE application. • Run the application. • Make security-related changes to configuration files. • Verify that strace does not log any changes made in the static configuration files.
Expected Test Results	The TOE writes security related configuration changes to /etc folder (for system-specific configuration) or in the user's home directory(for user-specific configuration) or /var/lib/ (for configurations controlled by UI and not intended to be directly modified by an administrator).
Pass/Fail with Explanation	Pass. The changes made in static client config file are not reflected in the strace logs.

7.1.29 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	NA. 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".

7.1.30 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	NA. The TSF does not provide any management functions. Note: An enterprise administrator manages the TOE via configuration files on each platform. There is no management CLI, GUI, or interface to manage the TOE.

7.1.31 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	NA. The TOE does not collect or transmit PII over a network

7.1.32 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:Linux...** The evaluator shall run the same application on two different Linux systems. The evaluator shall then compare their memory maps using pmap -x PID to ensure the two different instances share no mapping locations. **TD0798 has been applied**
Test Steps	• Start the application on two separate platforms . • Check status of the application to note the Process ID used in both the platforms. • Verify that the memory mappings on each platform does not occur.
Expected Test Results	No memory mapping locations should be placed at a consistent and explicit memory location
Pass/Fail with Explanation	Pass. The evaluator ran the application on two different machines and observed that the application running on two different machines share no memory mapping location. This meets testing requirements.

7.1.33 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:Linux...** The evaluator shall perform static analysis on the application to verify that both mmap is never be invoked with both the PROT_WRITE and PROT_EXEC permissions, and mprotect is never invoked with the PROT_EXEC permission.
Test Steps	• Use the strace command to perform static analysis on the application • Verify the mmap is never invoked with both PROT_WRITE and PROT_EXEC and mprotect is never invoked with PROT_EXEC permission
Expected Test Results	• mmap is never be invoked with both the PROT_WRITE and PROT_EXEC permissions. • mprotect is never invoked with the PROT_EXEC permission.
Pass/Fail with Explanation	Pass. The evaluator verified that no memory mapping requests are made with write and execute permissions. This meets the test requirements.

7.1.34 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:Linux...** The evaluator shall ensure that the application can successfully run on a system with either SELinux or AppArmor enabled and in enforce mode.
Test Steps	• Start the TOE application service. • Ensure that SE Linux is enabled and enforcing by executing the sestatus command.
Expected Test Results	The TOE runs successfully on a system that has SElinux enabled and enforcing in enforce mode.
Pass/Fail with Explanation	Pass. The TOE successfully run on a system that has SElinux enabled and enforcing in enforce mode. This meets the testing requirements.

7.1.35 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:Linux...** The evaluator shall run the program, mimicking normal usage, and note where all user-modifiable files are written. The evaluator shall ensure that there are no executable files stored in the same directories to which the application wrote user-modifiable files.
Test Steps	• Start the TOE application service. • List user-modifiable files opened by TOE application. • List the directories of the user-modifiable files from above step. • Verify that no executables are present in these listed directories. Note: By default, the installed directories containing user-modifiable files do not have executables in them – ST
Expected Test Results	There should be no executable files stored in the same directories to which the application write user-modifiable files.

Pass/Fail with Explanation	Pass. The user modifiable files are not written to the same directory containing executable files. This meets testing requirement.

7.1.36 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	• List the TOE ELF executables present on the system.
	• Run the python script cande.py for the above executables to detect stack-canary and verified that the **stack_chk_fail** flag is present.
	Note: Link for the python script
	https://github.com/commoncriteria/canary-detector/
	https://github.com/commoncriteria/canary-detector/blob/master/docs/Usage.md
	Here is the python script:

	cande.py
Expected Test Results	The TOE shall contain stack-based buffer overflow protection for ELF executables
Pass/Fail with Explanation	Pass. The stack-based buffer overflow protection is present in the TOE. This meets the testing requirements.

7.1.37 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	• List the Linux APIs used in the TOE as mentioned in the ST - Section 6 – TSS .
	 • Compare the above with the Platform Developer Webpage at:
	
	    o https://docs.oracle.com/javase/8/docs/api/
	
	    o https://javaee.github.io/javaee-spec/javadocs/
	
	    o https://github.com/corretto/corretto-8/blob/a6b2628f8074004f2c10bd7c276543a1acba412f
	
	    /src/jdk/src/share/classes/sun/security/x509/X500Name.java

Expected Test Results	TOE should support all the API listed in TSS.
Pass/Fail with Explanation	Pass. The evaluator verified that the APIs included in the ST are mentioned in the Platform Developer webpages. This meets the testing requirements.

7.1.38 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	• Start the TOE application. • Finding the enveil noarch file using command: sudo rpm -qa	grep enveil. • Displaying the information in the above file to indicate the version.

	• Note: SWID tags are not supported by TOE according to ST
Expected Test Results	The TOE version was verified in the installed RPM file.
Pass/Fail with Explanation	Pass. The TOE version was observed in the installation RPM file. This meets the testing requirements.

7.1.39 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	• Verify the status of the installed application. • Survey the installation directory for dynamic libraries. • Compare the listed libraries from the above command with those specified in ST and verify they match. Libraries specified in ST: • SEAL Homomorphic Encryption Library • GNU Multiple Precision Arithmetic Library • Java dependencies required for the JRE to execute
Expected Test Results	The TOE has dynamic libraries packaged with or employed by the application are limited to those in the assignment.
Pass/Fail with Explanation	Pass. It is verified that the dynamic libraries packaged with or employed by the application are limited to those in the assignment.

7.1.40 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	• Check for current version of application. • Check for available update using command in application documentation.

Item	Data
Expected Test Results	The TOE should check for update successfully without issuing an error.
Pass/Fail with Explanation	Pass. Application checks for new update, and the version being used is the documented version. This meets testing requirements.

7.1.41 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	- Start the TOE. - Check the version of the TOE. - Verify current version using the command in the section 3.5 Determining the Installed Version of ZeroReveal Client of the AGD matches the version mentioned in document (ST – Table 1 in Section 1.1 and).
Expected Test Results	The current version of TOE should match that of the documented and installed version.
Pass/Fail with Explanation	Pass. It is verified from the output of the above commands that the current version is the installed and documented version. This meets the testing requirements.

7.1.42 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. **Platforms:Apple iOS...** The evaluator shall consider the requirement met because the platform forces applications to write all data within the application working directory (sandbox). **For all other platforms**, the evaluator shall perform the following test: 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	- Install the TOE and confirm status. - Generate hashed copy of all executables. - Run the application. - Generate hashed copy of all executables. - Compare hashes before and after application is run.
Expected Test Results	Application executable files should not be changed when the application is run.
Pass/Fail with Explanation	Pass. Hash of all the executables files are verified to be identical before and after running the application. This meets testing requirements.

7.1.43 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,the evaluator shall verify that application updates are distributed in the format supported by the platform. This varies per platform:* **Platforms:Linux...** The evaluator shall ensure that the application is packaged in the format of the package management infrastructure of the chosen distribution. For example, applications running on Red Hat and Red Hat derivatives shall be packaged in RPM format. Applications running on Debian and Debian derivatives shall be packaged in DEB format. **TD0628 has been applied.**
Test Steps	- Inspect the TOE application package RPM file in a hex editor. - Verify that it is packaged in .rpm format. - HEX Signature List for rpm format: Source Wikipedia
Expected Test Results	Application updates are distributed in RPM format.
Pass/Fail with Explanation	Pass. The package is verified to be in RPM format. This meets the testing requirements.

7.1.44 FPT_TUD_EXT.2.2 Test #1

Item	Data
Test Assurance Activity	**Platforms:Android...** The evaluator shall consider the requirement met because the platform forces applications to write all data within the application working directory (sandbox). **Platforms:Apple iOS...** The evaluator shall consider the requirement met because the platform forces applications to write all data within the application working directory (sandbox). **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	• Record the path of every file on the entire filesystem before installing the TOE and save the output. • Install the TOE. • Start the application and confirm the application is running after configuration. • Uninstall the TOE. • Record the path of every file on the entire filesystem and save the output. • Verify that no files, other than configuration, output, and audit/log files, have been added to the filesystem.
Expected Test Results	No files other than the configuration, log and output file should be added to the system after uninstallation.
Pass/Fail with Explanation	Pass. No files other than configuration, output, and audit/log files that have been added to the filesystem

7.1.45 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	TOE as GUI server • Log into the TOE via GUI • Verify that user is successfully logged in to the TOE. • Verify the connection succeeds, and traffic is encrypted with TLS TOE as HTTPS server via REST-API • Configure the TOE as HTTPS server via REST-API • Establish a connection with the TOE over TLS • Verify the connection was successful • Verify the connection succeeds, and traffic is encrypted with TLS TOE as client to ZR Server • Configure the TOE to the TLS server • Establish a connection with the TOE over TLS using the cipher ECDHE-RSA-AES256-GCM-SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture TOE as client to LDAP server • Configure the TOE to the TLS server • Establish a connection with the TOE over TLS using the cipher ECDHE-RSA-AES256-GCM-SHA384 • Verify the connection was successful

Item	Data
	• Verify the connection was successful via packet capture
Expected Test Results	All the traffic captured when the TOE is exercised should be encrypted by either TLS or HTTPS
Pass/Fail with Explanation	Pass. The communication established between trusted IT product and TOE using HTTPS or TLS were encrypted. This meets the testing requirements.

7.1.46 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 done in conjunction with FTP_DIT_EXT.1.1 Test #1 where each Wireshark capture evidence was further analyzed to ensure that no sensitive data is transmitted as plain-text and was sent as encrypted application data for TLS connections and encrypted packets for HTTPS connections. This meets the testing requirements

7.1.47 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	NA. No credentials are transmitted. Mutually authenticated connection is established using certificates.

7.1.48 FTP_DIT_EXT.1.1 Test #4

Item	Data
Test Assurance Activity	**Platforms:Android...** If "**not transmit any data**" is selected, the evaluator shall ensure that the application's AndroidManifest.xml file does not contain a uses-permission or uses-permission-sdk-23 tag containing android:name="android.permission.INTERNET".

	In this case, it is not necessary to perform the above Tests 1, 2, or 3, as the platform will not allow the application to perform any network communication.
Pass/Fail with Explanation	N/A. The platfrom is Linux.

7.1.49 FTP_DIT_EXT.1.1 Test #5

Item	Data
Test Assurance Activity	**Platforms:Apple iOS...** If "**encrypt all transmitted data**" is selected, the evaluator shall ensure that the application's Info.plist file does not contain the NSAllowsArbitraryLoads or NSExceptionAllowsInsecureHTTPLoads keys, as these keys disable iOS's Application Transport Security feature.
Pass/Fail with Explanation	N/A. The platfrom is Linux.

7.2 PKG_TLSC (ZR Client to ZR Server)

7.2.1 FCS_TLSC_EXT.1.1 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	• Configure the TOE to the TLS server • Establish a connection with the TOE over TLS using the cipher ECDHE-RSA-AES256-GCM-SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture • Establish a connection with the TOE over TLS using the cipher ECDHE-ECDSA-AES256-GCM-SHA384 • Verify the connection was successful • Verify the connection was successful via packet capture

Expected Test Results	The TOE will make a connection with each of the supported ciphers
Pass/Fail with Explanation	Pass. The TOE was able to make each connection using the supported ciphersuites. This meets the test requirements.

7.2.2 FCS_TLSC_EXT.1.1 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	• Create a server certificate with the Server Authentication in the extendedKeyUsage field • Connection details from FCS_TLSC_EXT.2.1 Test 1 were used to show a successful TLS connection. • Verify the connection is established • Verify the connection is established via packet capture • Create a server certificate without the Server Authentication in the extendedKeyUsage field • Use Acumen-tlsc tool to initiate a connection to the TOE with a server certificate without the Server Authentication field resulted in a failure • Verify the connection failure via packet capture • Verify the connection failure is via logs
Expected Test Results	The TOE accepts the connection if the server certificate does contain the proper validation of extended key usage field.

	The TOE rejects the connection if the server certificate does not contain the proper validation of extended key usage field.
Pass/Fail with Explanation	Pass. The TOE rejects the connection with a server without a Server Authentication extended keyusage field. This meets the testing requirements.

7.2.3 FCS_TLSC_EXT.1.1 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	• Use Acumen-tlsc tool to initiate a connection to the TOE with an RSA certificate and ECDSA cipher suite resulted in a failure • Verify the connection failure via packet capture • Verify the connection failure is via logs
Expected Test Results	The TOE rejects the connection because the certificate does not match the server-selected ciphersuite.
Pass/Fail with Explanation	Pass. The TOE denied a connection to a server using a certificate that doesn't match the ciphersuite. This meets the testing requirements.

7.2.4 FCS_TLSC_EXT.1.1 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	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with the non-supported ciphersuite (NULL_WITH_NULL_NULL) • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection with an unsupported cipher suite TLS_NULL_WITH_NULL_NULL.
Pass/Fail with Explanation	Pass. The TOE denied the connection to a server using a NULL ciphersuite. This meets the testing requirements.

7.2.5 FCS_TLSC_EXT.1.1 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	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with the undefined TLS version • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	When the TLS version selected by the server in the Server Hello to a undefined TLS version, the TOE will not complete the connection.
Pass/Fail with Explanation	Pass. The TOE rejects the connection to a server using a undefined TLS version. This meets the testing requirements.

7.2.6 FCS_TLSC_EXT.1.1 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	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with the unsupported TLS version • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	When the TLS version selected by the server in the Server Hello to a non-supported TLS version, the TOE will not complete the connection.
Pass/Fail with Explanation	Pass. The TOE rejects the connection to a server using a non-supported TLS version. This meets the testing requirements.

7.2.7 FCS_TLSC_EXT.1.1 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.

Item	Data
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with the modified server nonce • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	When one byte in the server's nonce in the Server Hello handshake message is changed, the TOE the client rejects the Server Key Exchange handshake message.
Pass/Fail with Explanation	Pass. The TOE rejects the connection to a server with a modified nonce in the handshake message. This meets the testing requirements.

7.2.8 FCS_TLSC_EXT.1.1 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	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with an unsupported ciphersuite • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	When the server's selected ciphersuite in the Server Hello handshake message is modified to be a ciphersuite not presented in the Client Hello handshake message, the TOE does not complete the connection.
Pass/Fail with Explanation	Pass. When the server's selected ciphersuite in the Server Hello handshake message is modified to be a ciphersuite not presented in the Client Hello handshake message, the TOE does not complete the connection. This meets the testing requirements.

7.2.9 FCS_TLSC_EXT.1.1 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.
Test Steps	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with a modified signature in the Server Key Exchange

	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection after receiving a modified server certificate verify message
Pass/Fail with Explanation	Pass. The TOE rejects the connection after receiving a modified server certificate verify message. This meets the testing requirements.

7.2.10 FCS_TLSC_EXT.1.1 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	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with a modified Server Finished handshake message • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	When a byte in the Server Finished handshake message is modified, the TOE does not complete the connection.
Pass/Fail with Explanation	Pass. When a byte in the Server Finished handshake message is modified, the TOE does not complete the connection. This meets the test requirements.

7.2.11 FCS_TLSC_EXT.1.1 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	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with a garbled message after the Change Cipher Spec message is issued • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection when a byte in the Server Finished handshake message is modified
Pass/Fail with Explanation	Pass. When a garbled message is sent, the TOE rejects the connection. This meets the testing requirements.

7.2.12 FCS_TLSC_EXT.1.2 Test #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. Test 1: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails. Note that some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1. **TD0499 has been applied.**
Test Steps	CN as IP Address - TOE's settings and certificate details - Create a Server Certificate with no SAN and CN that does not match the reference identifier. - Use the Acumen-tlsc tool to initiate a connection to the TOE using a certificate missing the SAN but with a CN that does not match the reference identifier - Verify the connection failure via packet capture - Verify the connection failure via logs CN as FQDN - TOE's settings and certificate details - Create a Server Certificate with no SAN and CN that does not match the reference identifier. - Use the OpenSSL to initiate a connection to the TOE using a certificate missing the SAN but with a CN that does not match the reference identifier - Verify the connection failure

Item	Data
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	When a server certificate does not contain an identifier in either the Subject Alternative Name (SAN) or Common Name (CN) that matches the reference identifier, the TOE client rejects the Server Key Exchange handshake message.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when presented with a server certificate with Common Name (CN) that does not match the reference identifier and does not contain a Subject Alternative Name (SAN). This meets the testing requirements.

7.2.13 FCS_TLSC_EXT.1.2 Test #2

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 2: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type.
	TD0499 has been applied.
Test Steps	CN as IP Address
	• TOE's settings and certificate details
	• Create a Server Certificate with a SAN does not match the reference identifier and a CN that does match the reference identifier
	• Use the Acumen-tlsc tool to initiate a connection to the TOE using a certificate the matched CN but with a SAN that does not match the reference identifier
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
	CN as FQDN
	• TOE's settings and certificate details
	• Create a Server Certificate with a SAN does not match the reference identifier and a CN that does match the reference identifier

	• Use the OpenSSL to initiate a connection to the TOE using a certificate the matched CN but with a SAN that does not match the reference identifier • Verify the connection failure • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection when the reference identifier in the SAN does not match.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when presented with a server certificate with a Common Name (CN) that matches the reference identifier and a Subject Alternative Name (SAN) that does not match the reference identifier. This meets the testing requirements.

7.2.14 FCS_TLSC_EXT.1.2 Test #3

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 3: [conditional] If **the TOE does not mandate the presence of the SAN extension**, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted. **TD0499 has been applied.**
Pass/Fail with Explanation	N/A. The TOE mandates the presence of the SAN extension

7.2.15 FCS_TLSC_EXT.1.2 Test #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.

	Test 4: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. **TD0499 has been applied.**
Test Steps	CN as IP Address • TOE's settings and certificate details • Create a Server Certificate with a SAN that does match the reference identifier and a CN that does not match the reference identifier • Use the OpenSSL to initiate a connection to the TOE using a certificate the matched SAN but with a CN that does not match the reference identifier • Verify the connection succeeds • Verify the connection succeeds via packet capture CN as FQDN • TOE's settings and certificate details • Create a Server Certificate with a SAN that does match the reference identifier and a CN that does not match the reference identifier • Use the OpenSSL to initiate a connection to the TOE using a certificate the matched SAN but with a CN that does not match the reference identifier • Verify the connection succeeds • Verify the connection succeeds via packet capture
Expected Test Results	The TOE accepts the connection when a server certificate contains a CN that mismatches the reference identifier and does contains the SAN extension that matches.
Pass/Fail with Explanation	Pass. The TOE accepts the connection when presented with a server certificate with a CN that does not match the reference identifier and SAN extension that matches the reference identifier. This meets the testing requirements.

7.2.16 FCS_TLSC_EXT.1.2 Test #5.1

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 5.1: [conditional]: If **wildcards are supported**, the evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails. **TD0499 has been applied.**

Test Steps	CN as FQDN
	
	    • Configure the correct reference identifier in the TOE.
	
	    • Create a server certificate containing a wildcard that is not in the left-most label of CN.
	
	    • Use the OpenSSL to initiate a connection to the TOE using a certificate containing a wildcard that is not in the left-most label of CN
	
	    • Verify the connection failure
	
	    • Verify the connection failure via packet capture
	
	    • Verify the connection failure via logs
	
	SAN as FQDN
	
	    • Configure the correct reference identifier in the TOE.
	
	    • Create a server certificate containing a wildcard that is not in the left-most label of SAN.
	
	    • Use the OpenSSL to initiate a connection to the TOE using a certificate containing a wildcard that is not in the left-most label of SAN
	
	    • Verify the connection failure
	
	    • Verify the connection failure via packet capture
	
	    • Verify the connection failure via logs

Expected Test Results	The TOE rejects the connection when a server certificate contains a wildcard that is not in the left-most label of the presented identifier.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when a server certificate contains a wildcard that is not in the left-most label of the presented identifies. This meets the testing requirements.

7.2.17 FCS_TLSC_EXT.1.2 Test #5.2(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 5.2: [conditional]: If **wildcards are supported**, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com).
	- The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds.
	TD0499 has been applied.
Test Steps	CN:FQDN
	• Configure a single left-most label reference identifier in the TOE.
	• Create a server certificate containing a wildcard in the left-most identifier but not preceding the public suffix
	• Use the Acumen-tlsc tool to initiate a connection to the TOE with a server certificate containing a wildcard in the left-most identifier resulted in success
	• Verify the connection succeeds via packet capture
	SAN:FQDN
	• Configure a single left-most label reference identifier in the TOE.
	• Create a server certificate containing a wildcard in the left-most identifier but not preceding the public suffix
	• Use the Acumen-tlsc tool to initiate a connection to the TOE with a server certificate containing a a wildcard in the left-most identifier resulted in success
	• Verify the connection succeeds via packet capture
Expected Test Results	The TOE accepts the connection when the reference identifier is with a wildcard in the left most identifier.
Pass/Fail with Explanation	Pass. The TOE accepts the connection when a server certificate contains a wildcard in the left-most label of the presented identifier. This meets the testing requirements.

7.2.18 FCS_TLSC_EXT.1.2 Test #5.2(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 5.2: [conditional]: If **wildcards are supported**, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). - The evaluator shall configure the reference identifier 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	CN:FQDN • Configure no left-most label reference identifier in the TOE. • Create a server certificate containing a wildcard in the left-most identifier but not preceding the public suffix • Use the Acumen-tlsc tool to initiate a connection to the TOE with a server certificate containing a wildcard in the left-most identifier resulted in failure • Verify the connection failure via packet capture • Verify the connection failure via logs SAN:FQDN • Configure no left-most label reference identifier in the TOE. • Create a server certificate containing a wildcard in the left-most identifier but not preceding the public suffix • Use the Acumen-tlsc tool to initiate a connection to the TOE with a server certificate containing a a wildcard in the left-most identifier resulted in failure • Verify the connection failure via packet capture • Verify the connection failure via logs

Expected Test Results	The TOE rejects the connection when the reference identifier is with a wildcard in the left most identifier.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when a server certificate contains a wildcard in the left-most label of the presented identifiers. This meets the testing requirements.

7.2.19 FCS_TLSC_EXT.1.2 Test #5.2(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 5.2: [conditional]: If **wildcards are supported**, the evaluator shall present a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.example.com). - The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	CN:FQDN • Configure two left-most label reference identifier in the TOE. • Create a server certificate containing a wildcard in the left-most identifier but not preceding the public suffix • Use the Acumen-tlsc tool to initiate a connection to the TOE with a server certificate containing a wildcard in the left-most identifier resulted in failure • Verify the connection failure via packet capture • Verify the connection failure via logs SAN:FQDN • Configure no left-most label reference identifier in the TOE. • Create a server certificate containing a wildcard in the left-most identifier but not preceding the public suffix

Item	Data
	• Use the Acumen-tlsc tool to initiate a connection to the TOE with a server certificate containing a a wildcard in the left-most identifier resulted in failure • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection when the reference identifier is with a wildcard in the left most identifier.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when a server certificate contains a wildcard in the left-most label of the presented identifiers. This meets the testing requirements.

7.2.20 FCS_TLSC_EXT.1.2 Test #5.3(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 5.3: [conditional]: If **wildcards are supported**, the evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). - The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	CN:FQDN • Configure a single left-most label reference identifier in the TOE. • Create a server certificate containing a wildcard in the left-most identifier immediately preceding the public suffix • Use the Acumen-tlsc tool to initiate a connection to the TOE with a server certificate containing a wildcard in the left-most identifier resulted in failure • Verify the connection failure via packet capture • Verify the connection failure via logs SAN:FQDN

	• Configure a single left-most label reference identifier in the TOE. • Create a server certificate containing a wildcard in the left-most identifier immediately preceding the public suffix • Use the Acumen-tlsc tool to initiate a connection to the TOE with a server certificate containing a a wildcard in the left-most identifier resulted in failure • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection when the reference identifier is with a wildcard in the left most identifier.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when a server certificate contains a wildcard in the left-most label of the presented identifiers. This meets the testing requirements.

7.2.21 FCS_TLSC_EXT.1.2 Test #5.3(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 5.3: [conditional]: If **wildcards are supported**, the evaluator shall present a server certificate containing a wildcard in the left-most label immediately preceding the public suffix (e.g. *.com). - The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.com) and verify that the connection fails. **TD0499 has been applied.**
Test Steps	CN:FQDN • Configure two left-most label reference identifier in the TOE. • Create a server certificate containing a wildcard in the left-most identifier immediately preceding the public suffix • Use the Acumen-tlsc tool to initiate a connection to the TOE with a server certificate containing a wildcard in the left-most identifier resulted in failure • Verify the connection failure via packet capture

	• Verify the connection failure via logs SAN:FQDN • Configure two left-most label reference identifier in the TOE. • Create a server certificate containing a wildcard in the left-most identifier immediately preceding the public suffix • Use the Acumen-tlsc tool to initiate a connection to the TOE with a server certificate containing a a wildcard in the left-most identifier resulted in failure • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection when the reference identifier is with a wildcard in the left most identifier.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when a server certificate contains a wildcard in the left-most label of the presented identifiers. This meets the testing requirements.

7.2.22 FCS_TLSC_EXT.1.2 Test #5.4

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 5.4: [conditional]: If **wildcards are <u>not</u> 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. The TOE supports wildcards

7.2.23 FCS_TLSC_EXT.1.2 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: [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. TOE does not support URI or Service name reference identifiers are supported

7.2.24 FCS_TLSC_EXT.1.2 Test #7

Item	Data
Test Assurance Activity	Test 7: [conditional] If **pinned certificates are supported** the evaluator shall present a certificate that does not match the pinned certificate and verify that the connection fails.
Pass/Fail with Explanation	N/A. TOE does not support certificate pinning.

7.2.25 FCS_TLSC_EXT.1.3 Test #1a

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that a server using a certifcate with a valid certification path successfully connects. **TD0513 has been applied.**
Test Steps	• Create a full chain of certificates to connect to the TOE • Configure TOE to connect to the TLS server • Attempt the connection from the TOE to the TLS server

Item	Data
	• Verify the connection is successful • Verify the connection is successful via packet capture
Expected Test Results	• When a complete cert chain is presented, a TLS connection can be established
Pass/Fail with Explanation	Pass. When a complete certificate trust chain is present, the TOE can make a successful connection. This meets the test requirements.

7.2.26 FCS_TLSC_EXT.1.3 Test #1b

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 a full chain of certificates to connect to the TOE • Configure the correct chain on the TOE's trust store. • Create a certificate signed with an invalid certification path to TOE's trust store. • Attempt to connect to the TOE with the invalid intermediate CA and verify the connection failed • Verify that the connection failed via packet capture • Verify that the connection failed via logs
Expected Test Results	When an incomplete cert chain is present, a TLS connection cannot be established
Pass/Fail with Explanation	Pass. When an incomplete certificate trust chain is presented, the TOE is not able to make a successful connection. This meets the testing requirements.

7.2.27 FCS_TLSC_EXT.1.3 Test #1c

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.**
Test Steps	• Delete the ICA2 certificate from the chain from the TOE's truststore

	• Attempt to connect to the TOE without the intermediate CA and verify the connection failed
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	When an incomplete cert chain is present, a TLS connection cannot be established
Pass/Fail with Explanation	Pass. When an incomplete certificate trust chain is presented, the TOE is not able to make a successful connection. This meets the testing requirements.

7.2.28 FCS_TLSC_EXT.1.3 Test #2

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 2: The evaluator shall demonstrate that a server using a certificate which has been revoked results in an authentication failure.
Pass/Fail with Explanation	Pass. Test covered by FIA_X509_EXT.1.1/Rev Test #3 and FIA_X509_EXT.1.1/Rev Test #4. TOE rejects the connection with revoked certificates. This meets the testing requirements.

7.2.29 FCS_TLSC_EXT.1.3 Test #3

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 3: The evaluator shall demonstrate that a server using a certificate which has passed its expiration date results in an authentication failure.
Pass/Fail with Explanation	Pass. Test covered by FIA_X509_EXT.1.1/Rev Test #2. TOE rejects the connection with expired certificates. This meets the testing requirements.

7.2.30 FCS_TLSC_EXT.1.3 Test #4

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 4: 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 valid identifier on TOE. • Create a Server Certificate with an invalid the reference identifier

	• Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with an invalid the reference identifier
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection with an invalid reference identifier
Pass/Fail with Explanation	Pass. The TOE rejects the connection with an invalid reference identifier. This meets the testing requirement.

7.2.31 FCS_TLSC_EXT.2.1 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	• Use the Acumen-tlsc tool to initiate a connection to the TOE without the Server's Certificate Request and verify that the connection succeeds • Verify the connection succeeds via packet capture
Expected Test Results	The TOE should not send a Client Certificate message during the handshake.
Pass/Fail with Explanation	Pass. The TOE did not send any client certificate packets to the server that was not configured for mutual authentication. This meets testing requirements.

7.2.32 FCS_TLSC_EXT.2.1 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	• Use the Acumen-tlsc tool to initiate a connection to the TOE with the Server's Certificate Request and verify that the connection succeeds • Verify the connection succeeds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message via packet capture
Expected Test Results	The TOE should respond with a non-empty certificate message and certificate verify message.

Pass/Fail with Explanation	Pass. The TOE does not send an empty client certificate and sends a certificate verify message. This meets testing requirements.

7.2.33 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.
Test Steps	• Create Server certificate with SHA1 signature. • Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with an unsupported signature algorithm. • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection when the server uses an unsupported signature algorithm
Pass/Fail with Explanation	Pass. The TOE rejected the connection when the server's certificate contains an unsupported signature algorithm.

7.2.34 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.
Test Steps	• Create Server ECDSA certificate with SHA256 signature • Use the Acumen-tlsc tool to initiate a connection to the TOE and verify that the connection fails with an unsupported signature algorithm • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection when the server uses an unsupported signature algorithm
Pass/Fail with Explanation	Pass. TOE rejects the connection when certificate presented with unsupported signature algorithm. This meets the testing requirement.

7.2.35 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.
Pass/Fail with Explanation	N/A- The TOE does not support for Renegotiation.

7.2.36 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.
Pass/Fail with Explanation	N/A- The TOE does not support for Renegotiation.

7.2.37 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.
Pass/Fail with Explanation	N/A- The TOE does not support for Renegotiation.

7.2.38 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.
Test Steps	• Initiate a connection with the TOE over TLS using the curve secp384r1 • Verify the connection succeeds with secp384r1 • Verify with packet capture.

Item	Data
Expected Test Results	The TOE accepts the supported curves
Pass/Fail with Explanation	Pass. The TOE successfully completes a connection when each of the supported elliptic curves is used. This meets the test requirements.

7.3 PKG_TLSS (ZR Client to User)

7.3.1 FCS_TLSS_EXT.1.1 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	- Establish a connection between the REST API Client and TOE over TLS using the cipher ECDHE-RSA-AES256-GCM-SHA384 - Verify the connection was successful - Verify the connection was successful via packet capture - Establish a connection between the REST API Client and TOE over TLS using the cipher ECDHE-ECDSA-AES256-GCM-SHA384 - Verify the connection was successful - Verify the connection was successful via packet capture
Pass/Fail with Explanation	Pass. The TOE was able to make each connection using the supported ciphersuites. This meets the test requirements.

7.3.2 FCS_TLSS_EXT.1.1 Test #2

Item	Data
Test Assurance Activity	The evaluator shall send a Client Hello to the server with a list of cipher suites that does not contain any of the cipher suites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL cipher suite and verify that the server denies the connection.
Test Steps	Use the Acumen-tlss tool to initiate a connection to the TOE and verify that the connection fails with the non-supported ciphersuite

	o NULL_WITH_NULL_NULL
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
	• Use the Acumen-tlss tool to initiate a connection to the TOE and verify that the connection fails with the non-supported ciphersuite
	o RSA_WITH__NULL_MD5
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	The TOE will reject the NULL connection and non-claimed ciphersuites
Pass/Fail with Explanation	Pass. The TOE denied a connection due to unsupported and weak cipher. This meets the testing requirement.

7.3.3 FCS_TLSS_EXT.1.1 Test #3

Item	Data
Test Assurance Activity	If **RSA key exchange is used in one of the selected ciphersuites**, the evaluator shall use a client to send a properly constructed Key Exchange message with a modified EncryptedPreMasterSecret field during the TLS handshake. The evaluator shall verify that the handshake is not completed successfully and no application data flows.
Pass/Fail with Explanation	N/A – TOE only Supports ECDHE key exchange

7.3.4 FCS_TLSS_EXT.1.1 Test #4.1

TD0469 removes this test.

7.3.5 FCS_TLSS_EXT.1.1 Test #4.2

Item	Data
Test Assurance Activity	Modify a byte in the data of the client's Finished handshake message, and verify that the server rejects the connection and does not send any application data.
Test Steps	• Use the Acumen-tlss tool to initiate a connection to the TOE and verify the connection fails when a byte is modified in the client finished handshake.
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	The TOE should reject the connection after it receives the modified Client Finished handshake message.

Pass/Fail with Explanation	Pass. The TOE rejects the connection after receiving the modified Client Handshake message. This meets the testing requirement.

7.3.6 FCS_TLSS_EXT.1.1 Test #4.3i

Item	Data
Test Assurance Activity	Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption):
	[conditional]: If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test:
	If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test:
	a) The evaluator shall send a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket.
	b) The evaluator shall verify the server does not send a NewSessionTicket handshake message (at any point in the handshake).
	c) The evaluator shall verify the Server Hello message contains a zero-length session identifier or passes the following steps:
	Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID.
	d) The evaluator shall complete the TLS handshake and capture the SessionID from the ServerHello.
	e) The evaluator shall send a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d).

Item	Data
	f) The evaluator shall verify the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data. **TD0588 and TD0779 has been applied.**
Pass/Fail with Explanation	N/A – TOE supports **session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2)**

7.3.7 FCS_TLSS_EXT.1.1 Test #4.3ii

Item	Data
Test Assurance Activity	Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption): **[conditional]: If the TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2),** the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): a) The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246). b) The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data. **TD0588 and TD0779 has been applied.**
Test Steps	Part a: • Use the Acumen-tlss tool to initiate a connection to the TOE and verify the connection succeeds

	• Verify the connection successful via packet capture
	
	    o Verify a successful handshake and capture the TOE-generated session ID in the Server Hello message.
	
	    o Verify a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages.
	
	Part b:
	
	• Use the Acumen-tlss tool to initiate a connection to the TOE and verify the connection failure
	
	• Verify the connection failure via packet capture
	
	    o Verify the initiated handshake and capture the TOE-generated session ID in the Server Hello message.
	
	    o Verify within the same handshake an unencrypted fatal Alert message generated immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake
	
	    o Verify a new TLS connection with a Client Hello using the previously captured session ID
	
	    o Verify the server terminates the connection in some way that prevents the flow of application data

Expected Test Results	The TOE accepts previous session ID and responds with ServerHello containing the same SessionID. The TOE rejects the connections with the previous session ID with ClientHello containing the same SessionID.
Pass/Fail with Explanation	Pass. The TOE accepts previous session ID and responds with ServerHello containing the same SessionID. The TOE rejects the connections with the previous session ID with ClientHello containing the same SessionID. This meets the testing requirements.

7.3.8 FCS_TLSS_EXT.1.1 Test #4.3iii

Item	Data
Test Assurance Activity	Demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption):

	[conditional]: If the TOE supports session tickets according to RFC5077, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):
	a) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE successfully resumes the session in accordance with section 3.1 of RFC 5077.
	b) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data.
	c) The evaluator shall send the TSF a Client Hello with a SessionTicket extension, and observe the TSF responds with a Server Hello with an empty SessionTicket extension. The evaluator shall then send the TSF a invalid Finished message, and observe that the TSF terminates the session without sending a valid newTicket message.
	Note: if the TSF sends a newTicket message prior to terminating the session, the evaluator shall confirm the ticket is invalid by attempting to use the ticket to renew the session and observe that the TSF either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data.
	TD0588 and TD0779 has been applied.
Pass/Fail with Explanation	N/A – TOE does not supports **session tickets according to RFC5077**

7.3.9 FCS_TLSS_EXT.1.1 Test #4.4

Item	Data

Test Assurance Activity	Send a message consisting of random bytes from the client after the client has issued the ChangeCipherSpec message and verify that the server denies the connection.
Test Steps	• Use the Acumen-tlss tool to initiate a connection to the TOE and verify that the connection fails. • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE should reject the connection after it receives the modified packet
Pass/Fail with Explanation	Pass. When a Client Hello is received with modified after ChangeCipherSpec message, the TOE does not accept the connection. This meets the testing requirements

7.3.10 FCS_TLSS_EXT.1.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall send a Client Hello requesting a connection with version SSL 2.0 and verify that the server denies the connection. The evaluator shall repeat this test with SSL 3.0 and TLS 1.0, and TLS 1.1 if it is selected.
Test Steps	• Use Acumen-tlss tool to initiate a connection to the TOE and verify the connections fail for non-supported TLS versions • Verify the connection fails with SSL v2.0 • Verify the connection failure via packet capture • Verify the connection failure via logs • Verify the connection fails with SSL v3.0 • Verify the connection failure via packet capture • Verify the connection failure via logs • Verify the connection fails with TLS v1.0 • Verify the connection failure via packet capture • Verify the connection failure via logs • Verify the connection fails with TLS v1.1 • Verify the connection failure via packet capture • Verify the connection failure via logs • Verify the connection succeeds with TLS v1.2 • Verify the connection succeeds using packet capture
Expected Test Results	• The TOE rejects and logs the SSL v2.0 connection attempts.

	• The TOE rejects SSL v3.0 and logs the connection attempts.
	• The TOE rejects TLS v1.0 and logs the connection attempts.
	• The TOE rejects TLS v1.1 and logs the connection attempts.
	• The TOE accepts TLS v1.2 and logs the connection attempts
Pass/Fail **with** **Explanation**	Pass. TOE does not make the connection with the non-supported SSL and TLS version. This meets the testing requirements.

7.3.11 FCS_TLSS_EXT.1.3 Test #1

Item	Data
Test Assurance Activity	Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient.
	Test 1: [conditional] If **RSA-based key establishment is selected**, the evaluator shall configure the TOE with a certificate containing a supported RSA size and attempt a connection. The evaluator shall verify that the size used matches that which is configured and that the connection is successfully established. The evaluator shall repeat this test for each supported size of RSA-based key establishment.
	TD0739 has been applied
Pass/Fail **with** **Explanation**	N/A. TOE only supports ECDHE parameters using elliptic curves

7.3.12 FCS_TLSS_EXT.1.3 Test #2

Item	Data
Test Assurance Activity	Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient.
	[conditional] If **finite-field (i.e. non-EC) Diffie-Hellman ciphers are selected**, the evaluator shall attempt a connection using a Diffie-Hellman key exchange with a supported parameter size or supported group. The evaluator shall verify that the key agreement parameters in the Key Exchange message are the ones configured. The evaluator shall repeat this test for each supported parameter size or group.
Pass/Fail **with** **Explanation**	N/A. TOE only supports ECDHE parameters using elliptic curves

7.3.13 FCS_TLSS_EXT.1.3 Test #3

Item	Data
Test Assurance Activity	Note that this testing can be accomplished in conjunction with other testing activities. For each of the following tests, determining that the size matches the expected size is sufficient. [conditional] If **ECDHE ciphers are selected**, the evaluator shall attempt a connection using an ECDHE ciphersuite with a supported curve. The evaluator shall verify that the key agreement parameters in the Key Exchange message are the ones configured. The evaluator shall repeat this test for each supported elliptic curve.
Test Steps	• Initiate a connection with the TOE over TLS using the curve secp384r1 • Verify the connection succeeds with secp384r1 • Verify with packet capture.
Expected Test Results	• The TOE accepts the supported curves
Pass/Fail with Explanation	Pass. The TOE successfully completes a connection when each of the supported elliptic curves is used. This meets the test requirements.

7.3.14 FCS_TLSS_EXT.2.2 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure the server to send a certificate request to the client. The client shall send a certificate_list structure which has a length of zero. The evaluator shall verify that no sensitive application data flows prior to termination; if error messages are sent, the evaluator shall observe that an non-mutually authenticated channel is established, observe the data received by the test client to ensure only the error message indicated in the TSS is provided, and observe that the channel is then terminated. **TD0770 has been applied**
Test Steps	• Use Acumen-tlss tool to initiate a connection to the TOE with a certificate_list structure which has a length of zero and show the connection failed. • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE should reject the connection when the client tries to connect with the zero-length certificate
Pass/Fail with Explanation	Pass. The TOE rejects the connection when the client tries to connect with the zero-length certificate. This meets the testing requirement.

7.3.15 FCS_TLSS_EXT.2.2 Test #2

Item	Data
Test Assurance Activity	The evaluator shall configure the server to send a certificate request to the client. The client shall send no client certificate message, and instead send a client key exchange message in an attempt to continue the handshake. The client is required to respond to the certificate request message, even if the certificate message is empty. The evaluator shall verify that the handshake is not finished successfully and no application data flows. **TD0770 has been applied**
Test Steps	• Use Acumen-tlss tool to initiate a connection to the TOE without client certificate and show the connection failed. • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection if the client does not send a client certificate
Pass/Fail with Explanation	Pass. The TOE rejects an attempt to open a mutually authenticated TLS connection where the client does not send a certificate. This meets the testing requirements.

7.3.16 FCS_TLSS_EXT.2.2 Test #3

Item	Data
Test Assurance Activity	The evaluator shall configure the server to send a certificate request to the client without the supported_signature_algorithm used by the client's certificate. The evaluator shall attempt a connection using the client certificate and verify no sensitive application data flows prior to termination; if error messages are sent, the evaluator shall observe that an non-mutually authenticated channel is established, observe the data received by the test client to ensure only the error message indicated in the TSS is provided, and observe that the channel is then terminated. **TD0770 has been applied**
Test Steps	• Create a Client Certificate with the unsupported_signature_algorithm • Use Acumen-tlss tool to initiate a connection to the TOE with the unsupported_signature_algorithm and show the connection failed. • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection when the client uses an invalid signature algorithm

Pass/Fail with Explanation	Pass. The TOE rejects the TLS connection attempt from a client containing an unsupported signature algorithm. This meets testing requirements.

7.3.17 FCS_TLSS_EXT.2.2 Test #4

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. Using the administrative guidance, the evaluator shall then load a certificate or certificates 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, load the modified certificate path, and verify that no sensitive application data flows prior to termination; if error messages are sent, the evaluator shall observe that an non-mutually authenticated channel is established, observe the data received by the test client to ensure only the error message indicated in the TSS is provided, and observe that the channel is then terminated **TD0770 has been applied**
Test Steps	Valid certificate chain - Create a full chain of certificates to connect to the TOE. - Upload a complete certificate validation chain to the TOE. - Attempt to connect to the TOE with the full chain of proper certificates - Verify the connection is successful - Verify the connection is successful via packet capture Invalid certificate chain - Delete the ICA certificate from the chain - Use Acumen-tlss tool to initiate a connection to the TOE without a valid certification path and show the connection failed. - Verify the connection failure via packet capture - Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection if the client certificate was not issued by a trusted root CA.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when the entire certificate chain is not presented. This meets the test requirements.

7.3.18 FCS_TLSS_EXT.2.2 Test #5

Item	Data
Test Assurance Activity	The aim of this test is to check the response of the server when it receives a client identity certificate that is signed by an impostor CA (either Root CA or intermediate CA). To carry out this test the evaluator shall configure the client to send a client identity certificate with an issuer field that identifies a CA recognised by the TOE as a trusted CA, but where the key used for the signature on the client certificate does not in fact correspond to the CA certificate trusted by the TOE (meaning that the client certificate is invalid because its certification path does not in fact terminate in the claimed CA certificate). The evaluator shall verify that no sensitive application data flows prior to termination; if error messages are sent, the evaluator shall observe that an non-mutually authenticated channel is established, observe the data received by the test client to ensure only the error message indicated in the TSS is provided, and observe that the channel is then terminated. **TD0770 has been applied**
Test Steps	• TOE CA details • Create a CA certificate whose CN matches with the CA certificate on the TOE but with different key. • Use Acumen-tlss tool to initiate a connection to the TOE with a client certificate signed by impostor CA and show the connection failed. • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection using an impostor CA
Pass/Fail with Explanation	Pass. The TOE rejects a connection from a client using a certificate that is signed by an impostor CA and would fail to validate. This meets the testing requirements.

7.3.19 FCS_TLSS_EXT.2.2 Test #6

Item	Data
Test Assurance Activity	The evaluator shall configure the client to send a certificate with the Client Authentication purpose in the extendedKeyUsage field and verify that the server accepts the attempted connection. The evaluator shall repeat this test without the Client Authentication purpose and shall verify no sensitive application data flows prior to termination; if error messages are sent, the evaluator shall observe that an non-mutually authenticated channel is established, observe the data received by the test client to ensure only the error message indicated in the TSS is

	provided, and observe that the channel is then terminated. Ideally, the two certificates should be identical except for the Client Authentication purpose. **TD0770 has been applied**
Test Steps	• Create a client certificate with the Client Authentication in the extendedKeyUsage field • Connection details from FCS_TLSS_EXT.1.1 Test 1 were used to show a successful TLS connection. The console output shows a successful TLS connection • Verify the connection is established via packet capture • Create a server certificate without the Server Authentication in the extendedKeyUsage field • Use Acumen-tlss tool to initiate a connection to the TOE with a client certificate without the Client Authentication field resulted in a failure • Verify the connection failure via packet capture • Verify the connection failure is via logs
Expected Test Results	The TOE accepts the connection if the client certificate does contain the proper validation of extended key usage field. The TOE rejects the connection if the client certificate does not contain the proper validation of extended key usage field.
Pass/Fail with Explanation	Pass. The TOE rejects the connection with a client without a Client Authentication extended keyusage field. This meets the testing requirements.

7.3.20 FCS_TLSS_EXT.2.2 Test #7(a)

Item	Data
Test Assurance Activity	Configure the server to require mutual authentication and then modify a byte in the client's certificate. The evaluator shall verify that the server rejects the connection.
Test Steps	• Use Acumen-tlss tool to initiate a connection to the TOE with a modified client certificate resulted in a failure • Verify the connection failure via packet capture • Verify the connection failure is via logs
Expected Test Results	The TOE rejects the connection after receiving a modified client certificate
Pass/Fail with Explanation	Pass. The TOE rejects the TLS connection because of the modified certificate. This meets the testing requirements.

7.3.21 FCS_TLSS_EXT.2.2 Test #7(b)

Item	Data
Test Assurance Activity	Configure the server to require mutual authentication and then modify a byte in the signature block of the client's Certificate Verify handshake message. The evaluator shall verify that the server rejects the connection.
Test Steps	• Use Acumen-tlss tool to initiate a connection to the TOE with a client's modified Certificate Verify handshake message resulted in a failure • Verify the connection failure via packet capture • Verify the connection failure is via logs
Expected Test Results	The TOE rejects the connection after receiving a modified client certificate verify message
Pass/Fail with Explanation	Pass. The TOE rejects the connection after receiving the modified client's Certificate verify message. This meets the testing requirements.

7.3.22 FCS_TLSS_EXT.2.3 Test #1

Item	Data
Test Assurance Activity	The evaluator shall send a client certificate with an identifier that does not match any of the expected identifiers and verify that the server denies the connection. The matching itself might be performed outside the TOE (e.g. when passing the certificate on to a directory server for comparison).
Test Steps	• Configure the Client certificate with reference identifier not configured on TOE. • Use Acumen-tlss tool to initiate a connection to the TOE with a client's modified reference identifier resulted in a failure • Verify the connection failure via packet capture • Verify the connection failure is via logs
Expected Test Results	The TOE rejects a client connection if the client certificate does not contain a valid identifier
Pass/Fail with Explanation	Pass. The TOE rejects the connection with client certificate having identifier that does not match any of the expected identifiers. This meets the testing requirements.

7.3.23 FCS_TLSS_EXT.3.1 Test #1

Item	Data
Test Assurance Activity	The evaluator shall configure the server to send the signature_algorithms extension in the Certificate Request message indicating that the hash algorithm used by the client's certificate is not supported. The evaluator shall attempt a connection using that client certificate and verify that the server denies the client's connection.

Test Steps	• Create a Client Certificate with the unsupported_signature_algorithm
	• Use Acumen-tlss tool to initiate a connection to the TOE with the unsupported_signature_algorithm and show the connection failed.
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	The TOE rejects the connection when the client uses an invalid signature algorithm
Pass/Fail with Explanation	Pass. The TOE rejects the TLS connection attempt from a client containing an unsupported signature algorithm extension. This meets testing requirements.

7.3.24 FCS_TLSS_EXT.4.2 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 the "renegotiation_info" field is included in the ServerHello message.
Pass/Fail with Explanation	N/A – The TOE does not support for Renegotiation.

7.3.25 FCS_TLSS_EXT.4.2 Test #2

Item	Data
Test Assurance Activity	The evaluator shall modify the length portion of the field in the ClientHello message in the initial handshake to be non-zero and verify that the server sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.
Pass/Fail with Explanation	N/A – The TOE does not support for Renegotiation.

7.3.26 FCS_TLSS_EXT.4.2 Test #3

Item	Data
Test Assurance Activity	The evaluator shall modify the "client_verify_data" or "server_verify_data" value in the ClientHello message received during secure renegotiation and verify that the server terminates the connection.
Pass/Fail with Explanation	N/A – The TOE does not support for Renegotiation.

7.4 X509 (ZR Client to ZR Server)

7.4.1 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	• Establish 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.

- Configure the CA certificate lacking the basicConstraints extension.

- Verify that the signing CA certificate does not contain the basicConstraints extension

- Sign the certificate using CA certificate does not contain the basicConstraints extension.

- Use Acumen-tlsc tool to initiate a connection to the TOE with the chain does not contain the basicConstraints extension and verify the connections failed

- Verify the connection failure via packet capture

- Verify the connection failure via logs

- By setting the basicConstraints field in an issuing certificate to have CA=False.

    - Configure the CA certificate with the flag in the basicConstraints extension set to FALSE.

    - Verify that the signing CA certificate has the cA flag in the basicConstraints extension set to FALSE

    - Sign the certificate using ICA with basic constraints set to FALSE.

    - Use Acumen-tlsc tool to initiate a connection to the TOE with the chain with the basicConstraints set to FALSE and verify the connections failed

    - Verify the connection failure via packet capture

    - Verify the connection failure via logs

- By omitting the CA signing bit of the key usage field in an issuing certificate.

    - Configure the CA certificate lacking the CA signing bit in the Key usage field.

    - Load the certificate lacking the CA signing bit on the TLS server.

    - Sign the certificate using ICA with no certificate sign key usage.

    - Use Acumen-tlsc tool to initiate a connection to the TOE without the CA signing bit of the key usage field in the chain and verify the connections failed

	- Verify the connection failure via packet capture - Verify the connection failure via logs - By setting the path length field of a valid CA field to a value strictly less than the certificate path. - Configure the root CA certificate with the Path length of 1. - Configure the Intermediate CA1 certificate with the Path length of 0. - Configure the Intermediate CA2 certificate with the Path length of 0. - Sign the node certificate with ICA2. - Use Acumen-tlsc tool to initiate a connection to the TOE with path length field of a valid CA field to a value strictly less than the certificate path and verify the connections failed - Verify the connection failure via packet capture - Verify the connection failure via logs Valid certificate chain - Create a full chain of certificates to connect to the TOE. - Upload a complete certificate validation chain to the TOE. - Attempt to connect to the TOE with the full chain of proper certificates - Verify the connection is successful - Verify the connection is successful via packet capture Invalid certificate chain - Delete the ICA2 certificate from the chain - Attempt to connect to the TOE without the intermediate CA and verify the connection failed - Verify the connection failure via packet capture - Verify the connection failure via logs
Expected Test Results	When a complete cert chain is present, a TLS connection can be established

Item	Data
	• When an incomplete cert chain is present, a TLS connection cannot be established
Pass/Fail with Explanation	Pass. TOE only makes the connection when the valid certificate chain exists on the device. This meets the testing requirement.

7.4.2 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 | • Create a certificate that is expired according to the TOE

• Use Acumen-tlsc tool to initiate a connection to the TOE with the expired certificate and verify the connections failed

• Verify the connection failure via packet capture

Verify the connection failure via logs |
| Expected Test Results | The TOE rejects the connection with an expired certificate. |
| Pass/Fail with Explanation | Pass. The TOE rejects a connection with an expired certificate. This meets the testing requirement. |

7.4.3 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.
	o 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	CRL
	• Create chain of certificates with CRL Extended Key Usage.
	• Make sure that TOE is configured for CRL
	• Attempt connection with valid certificate
	• Verify the successful connection
	• Verify the successful connection via packet capture.
	• Revoke the peer leaf certificate
	• Attempt to make a connection and verify the connection failed

	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
	• Unrevoke the peer's leaf certificate and revoke the server's intermediate certificate
	• Attempt to make a connection and verify the connection failed
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	• When the intermediate and leaf certificate is revoked the session will not be established
	• When the intermediate and leaf certificate is not revoked the session will be established
Pass/Fail with Explanation	Pass. TOE rejects connection with revoked certificates. This meets the testing requirements.

7.4.4 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 configure the TSF to reject certificates if it cannot access valid status information, if so configurable. Then 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 and 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	• Create the CA signing the CRL to use a signing certificate that does not have the cRLsign key usage bit set • Generate new CRL with referenced to above certificate which doesn't have CRL sign • Import the invalid CRL into server • Make sure that TOE is configured for CRL • Attempt a connection with the server and verify the connection failed • Verify the connection failure for failure via logs • Verify the connection failure via packet capture
Expected Test Results	The TOE does not validate the CRL when CA signing the CRL to use a signing certificate that does not have the CRL sign key usage bit set.
Pass/Fail with Explanation	Pass. TOE rejects connection when CA signing the CRL does not have the CRLsign key usage bit. This meets the testing requirements

7.4.5 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	• Use Acumen-tlsc tool to initiate a connection to the TOE modifying a byte in the first 8 bytes of the certificate and verify the connections failed • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects connections when a byte in the first 8 bytes of the certificate is modified.
Pass/Fail with Explanation	Pass. The TOE rejects connections when a byte in the first 8 bytes of the certificate is modified. This meets the testing requirements.

7.4.6 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	• Use Acumen-tlsc tool to initiate a connection to the TOE modifying the last byte of the certificate (part of the signature) and verify the connections failed • Verify the connection failure via packet capture Verify the connection failure via logs
Expected Test Results	The TOE rejects connections when the last byte of the certificate is modified.
Pass/Fail with Explanation	Pass. The TOE rejects connections when the last byte of the certificate is modified. This meets the testing requirements.

7.4.7 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	• Use Acumen-tlsc tool to initiate a connection to the TOE modifying the public key of the certificate and verify the connections failed • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects connections when the public key of the certificate is modified.

Pass/Fail with Explanation	Pass. The TOE rejects connections when the public key of the certificate is modified. This meets the testing requirements.

7.4.8 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. |
| **Test Steps** | • Create a full chain of certificates to connect to the TOE.

• Upload a complete certificate validation chain to the TOE.

• Use OpenSSL to initiate a connection to the TOE with the full chain of proper certificates and verify the successful connection

• Verify the connection is successful via packet capture |
| **Expected Test Results** | When a complete cert chain is present, a TLS connection can be established. |
| **Pass/Fail with Explanation** | Pass. When a complete EC certificate trust chain is presented, the TOE is able to make a successful connection. This meets the test requirements |

7.4.9 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.
Test Steps	• Replace the ICA2 in the earlier test with a modified intermediate certificate with a named curve with an explicit format in the public key information field • Intermediate CA before modification • Modifying ICA using x509-mod tool • Intermediate CA after modification • Add modified certificate to certificate chain (Concatenate the Modified Intermediate CA and the root CA). • Use OpenSSL to initiate a connection to the TOE with the modified ICA2 certificate chain and verify the connection failed • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	When an incomplete cert chain is present, a TLS connection cannot be established
Pass/Fail with Explanation	Pass. When the public key information is modified in the intermediate certificate, TOE is unable to make the successful connection. This meets the test requirements.

7.4.10 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.
Test Steps	• Configure the CA certificate lacking the basicConstraints extension.
	• Verify that the signing CA certificate does not contain the basicConstraints extension
	• Sign the certificate using CA certificate does not contain the basicConstraints extension.
	• Use Acumen-tlsc tool to initiate a connection to the TOE with the chain does not contain the basicConstraints extension and verify the connections failed
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	The TOE rejects connections with the CA that does not contain the basicConstraints extension
Pass/Fail with Explanation	Pass. The TOE rejects certificates signed by a CA that does not contain the basicConstraints extension. This meets the testing requirements.

7.4.11 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
Test Steps	• Configure the CA certificate with the flag in the basicConstraints extension set to FALSE.
	• Verify that the signing CA certificate has the cA flag in the basicConstraints extension set to FALSE
	• Sign the certificate using ICA with basic constraints set to FALSE.
	• Use Acumen-tlsc tool to initiate a connection to the TOE with the chain with the basicConstraints set to FALSE and verify the connections failed
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	The TOE rejects connections with the CA with the basicConstraints extension set to FALSE
Pass/Fail with Explanation	Pass. The TOE rejects certificates signed by a CA that has the cA flag in the basicConstraints extension set to FALSE. This meets the testing requirements.

7.4.12 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	• Create chain of certificates with CRL Extended Key Usage. • Make sure that TOE is configured for CRL • Attempt connection with valid certificate • Verify the successful connection • Verify the successful connection via packet capture. • On the server, delete the ICA2 crl • Attempt to make a connection and verify the connection failed • Verify that the connection failure via packet capture • Verify that the connection failure via logs
Expected Test Results	The TOE rejects a connection when the CRL is not found
Pass/Fail with Explanation	Pass. The TOE makes the successful connection with the server (user) when certificate validity is confirmed and denies connection when the revocation status of the server certificate cannot be verified. This meets the testing requirements

7.4.13 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 covered by FIA_X509_EXT.1 Test#3 and FIA_X509_EXT.1 Test#4. The connection is rejected when an invalid certificate is presented. This meets the testing requirements.

7.5 X509 (ZR Client to User)

7.5.1 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** | - Establish 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. |

- Configure the CA certificate lacking the basicConstraints extension.

- Verify that the signing CA certificate does not contain the basicConstraints extension

- Sign the certificate using CA certificate does not contain the basicConstraints extension.

- Use Acumen-tlss tool to initiate a connection to the TOE with the chain does not contain the basicConstraints extension and verify the connections failed

- Verify the connection failure via packet capture

- Verify the connection failure via logs

- By setting the basicConstraints field in an issuing certificate to have CA=False.

  - Configure the CA certificate with the flag in the basicConstraints extension set to FALSE.

  - Verify that the signing CA certificate has the cA flag in the basicConstraints extension set to FALSE

  - Sign the certificate using ICA with basic constraints set to FALSE.

  - Use Acumen-tlss tool to initiate a connection to the TOE with the chain with the basicConstraints set to FALSE and verify the connections failed

  - Verify the connection failure via packet capture

  - Verify the connection failure via logs

- By omitting the CA signing bit of the key usage field in an issuing certificate.

  - Configure the CA certificate lacking the CA signing bit in the Key usage field.

  - Load the certificate lacking the CA signing bit on the TLS server.

  - Sign the certificate using ICA with no certificate sign key usage.

  - Use Acumen-tlss tool to initiate a connection to the TOE without the CA signing bit of the key usage field in the chain and verify the connections failed

  - Verify the connection failure via packet capture

	• Verify the connection failure via logs
	• By setting the path length field of a valid CA field to a value strictly less than the certificate path.
	• Configure the root CA certificate with the Path length of 1.
	• Configure the Intermediate CA1 certificate with the Path length of 0.
	• Configure the Intermediate CA2 certificate with the Path length of 0.
	• Sign the node certificate with ICA2.
	• Use Acumen-tlss tool to initiate a connection to the TOE with path length field of a valid CA field to a value strictly less than the certificate path and verify the connections failed
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
	Valid certificate chain
	• Create a full chain of certificates to connect to the TOE.
	• Upload a complete certificate validation chain to the TOE.
	• Attempt to connect to the TOE with the full chain of proper certificates
	• Verify the connection is successful
	• Verify the connection is successful via packet capture
	Invalid certificate chain
	• Delete the ICA2 certificate from the chain
	• Attempt to connect to the TOE without the intermediate CA and verify the connection failed
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	• When a complete cert chain is present, a TLS connection can be established • When an incomplete cert chain is present, a TLS connection cannot be established

Pass/Fail with Explanation	Pass. TOE only makes the connection when the valid certificate chain exists on the device. This meets the testing requirement.

7.5.2 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** | • Create a certificate that is expired according to the TOE

• Use Acumen-tlss tool to initiate a connection to the TOE with the expired certificate and verify the connections failed

• Verify the connection failure via packet capture

• Verify the connection failure via logs |
| **Expected Test Results** | The TOE rejects the connection with an expired certificate. |
| **Pass/Fail with Explanation** | Pass. The TOE rejects a coonection with an expired certificate. This meets the testing requirement. |

7.5.3 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.
	
	     o 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	CRL
	
	   • Create chain of certificates with CRL Extended Key Usage.
	
	   • Make sure that TOE is configured for CRL
	
	   • Attempt connection with valid certificate
	
	   • Verify the successful connection
	
	   • Verify the successful connection via packet capture.
	
	   • Revoke the peer leaf certificate
	
	   • Attempt to make a connection
	
	   • Verify the connection failure
	
	   • Verify the connection failure via packet capture

Item	Data
	• Verify the connection failure via logs
	• Unrevoke the peer's leaf certificate and revoke the client's intermediate certificate
	• Attempt to make a connection
	• Verify the connection failure
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	• When the intermediate and leaf certificate is revoked the session will not be established • When the intermediate and leaf certificate is not revoked the session will be established
Pass/Fail with Explanation	Pass. TOE rejects connection with revoked certificates. This meets the testing requirements.

7.5.4 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 configure the TSF to reject certificates if it cannot access valid status information, if so configurable. Then 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 and 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	• Create the CA signing the CRL to use a signing certificate that does not have the cRLsign key usage bit set • Generate new CRL with referenced to above certificate which doesn't have CRL sign • Import the invalid CRL into server • Make sure that TOE is configured for CRL • Attempt a connection with the server and verify the connection failed • Verify the connection failure for failure via logs • Verify the connection failure via packet capture
Expected Test Results	The TOE does not validate the CRL when CA signing the CRL to use a signing certificate that does not have the CRL sign key usage bit set.
Pass/Fail with Explanation	Pass. TOE rejects connection when CA signing the CRL does not have the CRLsign key usage bit. This meets the testing requirements

7.5.5 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	• Use Acumen-tlss tool to initiate a connection to the TOE modifying a byte in the first 8 bytes of the certificate and verify the connections failed • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects connections when a byte in the first 8 bytes of the certificate is modified.
Pass/Fail with Explanation	Pass. The TOE rejects connections when a byte in the first 8 bytes of the certificate is modified. This meets the testing requirements.

7.5.6 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	• Use Acumen-tlss tool to initiate a connection to the TOE modifying the last byte of the certificate (part of the signature) and verify the connections failed

Item	Data
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	The TOE rejects connections when the last byte of the certificate is modified.
Pass/Fail with Explanation	Pass. The TOE rejects connections when the last byte of the certificate is modified. This meets the testing requirements.

7.5.7 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	• Use Acumen-tlss tool to initiate a connection to the TOE modifying the public key of the certificate and verify the connections failed
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	The TOE rejects connections when the public key of the certificate is modified.
Pass/Fail with Explanation	Pass. The TOE rejects connections when the public key of the certificate is modified. This meets the testing requirements.

7.5.8 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.
Test Steps	• Create a full chain of certificates to connect to the TOE.
	• Upload a complete certificate validation chain to the TOE.
	• Use Acumen-tlss tool to initiate a connection to the TOE with the full chain of proper certificates and verify the successful connection
	• Verify the connection is successful via packet capture
Expected Test Results	When a complete cert chain is present, a TLS connection can be established.
Pass/Fail with Explanation	Pass. When a complete EC certificate trust chain is presented, the TOE is able to make a successful connection. This meets the test requirements

7.5.9 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.
Test Steps	• Replace the ICA2 in the earlier test with a modified intermediate certificate with a named curve with an explicit format in the public key information field • Intermediate CA before modification • Modifying ICA using x509-mod tool • Intermediate CA after modification • Add modified certificate to certificate chain (Concatenate the Modified Intermediate CA and the root CA). • Use openssl to initiate a connection to the TOE with the modified ICA2 certificate chain and verify the connection failed • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	When an incomplete cert chain is present, a TLS connection cannot be established
Pass/Fail with Explanation	Pass. When the public key information is modified in the intermediate certificate, TOE is unable to make the successful connection. This meets the test requirements.

7.5.10 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:
	(iii) as part of the validation of the peer certificate belonging to this chain; and/or
	(iv) when attempting to add the CA certificate without the basicConstraints extension to the TOE's trust store.
Test Steps	• Configure the CA certificate lacking the basicConstraints extension.
	• Verify that the signing CA certificate does not contain the basicConstraints extension
	• Sign the certificate using CA certificate does not contain the basicConstraints extension.
	• Use Acumen-tlss tool to initiate a connection to the TOE with the chain does not contain the basicConstraints extension and verify the connections failed
	• Verify the connection failure via packet capture
	• Verify the connection failure via logs
Expected Test Results	The TOE rejects connections with the CA that does not contain the basicConstraints extension
Pass/Fail with Explanation	Pass. The TOE rejects certificates signed by a CA that does not contain the basicConstraints extension. This meets the testing requirements.

7.5.11 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

(iii)   as part of the validation of the peer certificate belonging to this chain; and/or

(iv)   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

Test Steps	• Configure the CA certificate with the flag in the basicConstraints extension set to FALSE. • Verify that the signing CA certificate has the cA flag in the basicConstraints extension set to FALSE • Sign the certificate using ICA with basic constraints set to FALSE. • Use Acumen-tlss tool to initiate a connection to the TOE with the chain with the basicConstraints set to FALSE and verify the connections failed • Verify the connection failure via packet capture • Verify the connection failure via logs
Expected Test Results	The TOE rejects connections with the CA with the basicConstraints extension set to FALSE
Pass/Fail with Explanation	Pass. The TOE rejects certificates signed by a CA that has the cA flag in the basicConstraints extension set to FALSE. This meets the testing requirements.

7.5.12 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	• Create chain of certificates with CRL Extended Key Usage. • Make sure that TOE is configured for CRL • Attempt connection with valid certificate • Verify the successful connection • Verify the successful connection via packet capture. • On the server, delete the ICA2 crl • Attempt to make a connection • Verify that the connection failure • Verify that the connection failure via packet capture • Verify that the connection failure via logs
Expected Test Results	The TOE rejects a connection when the CRL is not found
Pass/Fail with Explanation	Pass.The TOE makes the successful connection with the client (user) when certificate validity is confirmed and denies connection when the revocation status of the client certificate cannot be verified. This meets the testing requirements

7.5.13 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 covered by FIA_X509_EXT.1 Test#3 and FIA_X509_EXT.1 Test#4.The connection is rejected when an invalid certificate is presented. This meets the testing requirements.

intertek
acumen
security

8   Conclusion

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

A. Appendix: CAVP Mapping

Algorithm	Standard	Modes Supported	CAVP Certificate
Cryptographic Asymmetric Key Generation (FCS_CKM.1/AK)			
RSA KeyGen	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	2048 bits and 3072 bits and greater	A4651
ECC KeyGen	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	Curves P-256 and P-384	A4651
Cryptographic Key Establishment (FCS_CKM.2)			
ECDHE Key Establishment	NIST SP 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	Curves P-384	A4651
Cryptographic Operation – Hashing (FCS_COP.1/Hash)			
SHA2-256	FIPS Pub 180-4	Digest size 256 bits	A4651
SHA2-384	FIPS Pub 180-4	Digest size 384 bits	A4651
SHA2-512	FIPS Pub 180-4	Digest size 512 bits	A4651
Cryptographic Operation – Keyed-Hash Message Authentication (FCS_COP.1/KeyedHash)			
HMAC-SHA2-256	FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'	Key size 256 bits, block size 512 bits, digest size 256 bits	A4651
HMAC-SHA2-384	FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'	Key size 384 bits, block size 512 bits, digest size 384 bits	A4651
HMAC-SHA2-512	FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code' and FIPS Pub 180-4 'Secure Hash Standard'	Key size 512 bits, block size 512 bits, digest size 512 bits	A4651
Cryptographic Operation – Signing (FCS_COP.1/Sig)			
RSA	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.	2048-bit or greater	A4651
ECDSA	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6.	P-256, P-384,	A4651
Cryptographic Operation - Encryption/Decryption (FCS_COP.1/SKC)			
AES-CCM	NIST SP 800-38C	256 bits	A4651
AES-GCM	NIST SP 800-38D	256 bits	A4651
Random Bit Generation from Application (FCS_RBG_EXT.2)			
HMAC_DRBG	NIST SP 800-90A	AES-256	A4651

End of Document