Assurance Activity Report for CACI Archon OS v3.0.0.2

AAR Version 1.6, July 11, 2024

CACI Archon OS v3.0.0.2 Security Target, Version 1.5

Protection Profile for General Purpose Operating Systems, Version 4.3
Functional Package for Transport Layer Security (TLS), Version 1.1

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:
CACI,

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

The TOE Evaluation was Sponsored by:
CACI,

Evaluation Personnel:
Fathi Nasraoui
Chaitanya Muzumdar
Snehal Gaonkar
Theo Ajibade

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
1.0	29/12/2023	Initial Release
1.1	16/01/2024	QA review
1.2	07/05/2024	Applying new TDs
1.3	09/06/2024	Release of new ST version
1.4	10/06/2024	Deleting some unrelated TSS and AGD activities
1.5	04/07/2024	Addressing ECR comments
1.6	11/07/2024	Addressing ECR comments

Table of Contents

1 TOE Overview

Archon OS v3.0.0.2 is an operating system (OS) based on Red Hat Enterprise Linux (RHEL) v8.10 that supports multiple users, user permissions, access controls, and cryptographic functionality.

Archon OS is an ostree-based packaging of Red Hat Enterprise Linux (RHEL), tailored for deployment on End User Devices (EUDs) specifically designed for Commercial Solutions for Classified (CSfC) solutions. The Archon OS ostree incorporates unmodified versions of the RHEL RPMs. Archon OS is curated to incorporate solely the essential OS options and applications pertinent to EUD functionality, with non-applicable components deliberately excluded.

1.1 TOE Description

1.1.1 Type

The TOE is a general-purpose operating system (OS), that supports multiple users, user permissions, access controls, and cryptographic functionality.

1.1.2 Evaluated Configuration

The TOE is a software TOE and has been evaluated on the following host platforms.

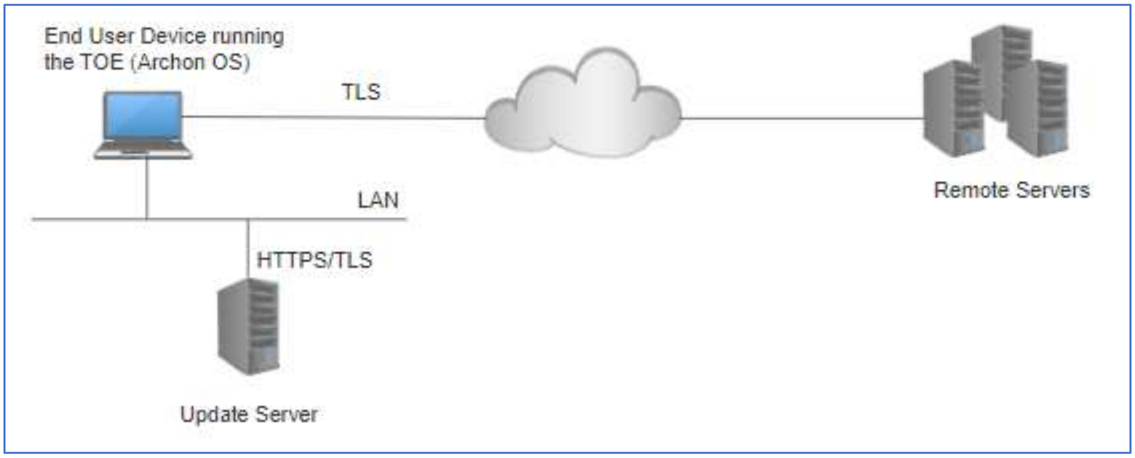
Table 1 – Archon OS v3.0.0.2 Hardware Platforms

Vendor	Model	CPU	CPU Microarchitecture	CPU Family
Dell Inc.	Latitude 5400	Intel® Core™ i5-8365U	Skylake	Whiskey Lake
	Latitude 5410	Intel® Core™ i7-10810U	Skylake	Comet Lake
	Latitude 5430	Intel® Core™ i7-1255U	Golden Cove	Alder Lake
	Precision 3260	Intel® Core™ i7-12700	Golden Cove	Alder Lake
	Precision 3570	Intel® Core™ i7-1255U	Golden Cove	Alder Lake
	Latitude 5440	Intel® Core™ i5-1335U	Raptor Cove	Raptor Lake
	Latitude 5540	Intel® Core™ i5-1335U	Raptor Cove	Raptor Lake
	Precision 3580	Intel® Core™ i5-1350P	Raptor Cove	Raptor Lake

1.1.3 Physical Boundary

The diagram below depicts a representative TOE deployment.

Figure 1: Representative TOE Deployment



The following items are required for the operational environment.

Table 2: Hardware and Software Environmental Components

Components	**Mandatory/ Optional**	**Description**
TOE Host	Mandatory	The hardware running the TOE. One of the systems listed in Table 1 above.
Update Server	Mandatory	Provides the ability to check for TOE software updates as well as providing signed updates. The TOE communicates with the Update Server using HTTPS over TLS.
Remote Servers	Mandatory	Servers that support multiple applications and provide multiple services.

2 Assurance Activities Identification

The Assurance Activities contained within this document include all those defined within the PP_OS_V4.3 and PKG_TLS_V1.1 based upon the core SFRs and those implemented based on selections within the PPs/PKGs.

3 Technical Decisions

All NIAP TDs issued to date and applicable to the PP and Functional Packages have been considered. The following table identifies all applicable TDs.

Table 3: Relevant Technical Decisions

Technical Decision	Applicable (Yes/No)	Exclusion Rationale (if applicable) and Notes
GPOS PP v4.3		
TD0839: Clarification for Local Administration in FTP_TRP.1.3	Yes	Modifies FTP_TRP.1.3 SFR, Application Note, TSS, AGD, and Test.
TD0821: Corrections to ECD for PP_OS_V4.3	Yes	
TD0812: Updated CC Conformance Claims in PP_OS_V4.3	Yes	
TD0809: Update to FCS_COP.1/SIGN for CNSA 1.0 compliance with secure Boot Exception	Yes	Modifies FCS_COP.1/SIGN SFR, TSS, and AGD. **Archives TD0727**
TD0789: Correction to TLS Selection in FIA_X509_EXT.2.1	Yes	Modifies FIA_X509_EXT.2.1 SFR and Test. Modifies FTP_ITC_EXT.1.1 SFR, Application Note, and Test.
TD0773: Updates to FIA_X509_EXT.1 for Exception Processing and Test Conditions	Yes	Modifies FIA_X509_EXT.1.1 Application Note, TSS, and Test. **Archives TD0692**
TD0713: Functional Package SFR mappings to objectives	Yes	
TD0712: Support for Bluetooth Standard 5.3	Yes	Modifies FCS_CKM.1 SFR, Application Note, TSS, AGD, and Test. Modifies FCS_COP.1/ENCRYPT SFR, Application Note, TSS, and Test.
TD0701: Incomplete selection reference in FCS_CM_EXT.4 TSS activities	Yes	Applies to FCS_CKM_EXT.4 TSS AA.
TD0696: Removal of 160-bit selection from FCS_COP.1/HASH & FCS_COP.1/KEYMAC	Yes	Modifies FCS_COP.1/HASH and FCS_COP.1.1/KEYHMAC SFRs.
TD0693: Typos in OSPP 4.3	Yes	Applies to FMT_MOF_EXT.1 Application Note and FMT_SMF_EXT.1 Application Note. Applies to FAU_GEN.1 Test.
TD0691: OSPP 4.3 Conditional authentication testing	Yes	Applies to FIA_AFL.1 Application Note and Test.
TD0675: Make FPT_W^X_EXT.1 Optional	Yes	
TLS Pkg v1.1		
TD0779: Updated Session Resumption Support in TLS package V1.1	Yes	The ST does not claim TLS server, however the TD **Archives TD0588** and therefore applies to this evaluation.
TD0770: TLSS.2 connection with no client cert	No	The ST does not claim TLS server.
TD0739: PKG_TLS_V1.1 has 2 different publication dates	Yes	The TD modifies FCS_TLSS_EXT.1.3 Test which doesn't apply to this evaluation, but also

Technical Decision	Applicable (Yes/No)	Exclusion Rationale (if applicable) and Notes
		mentions the two different dates for PKG_TLS_V1.1 (https and pdf) and that 03.01.2019 should be used and therefore the TD applies to this evaluation.
TD0726: Correction to (D) TLSS SFRs in TLS 1.1 FP	No	The ST does not claim TLS or DTLS server.
TD0513: CA Certificate loading	Yes	Applies to FCS_TLSC_EXT.1.3 Test.
TD0499: Testing with pinned certificates	Yes	Applies to FCS_TLSC_EXT.1.2 Test.
TD0469: Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	No	The ST does not claim TLS server.
TD0442: Updated TLS Ciphersuites for TLS Package	Yes	Modifies FCS_TLSC_EXT.1.1 SFR.

4 Test TOE’s Platform Equivalency

Based on the equivalency analysis done in the *Equivalency Analysis for Archon OS v3.0.0.0*, v2.1 the following table lists the TOE’s equivalent models.

Table 4: Hardware Appliances

TOE’s Model	TOE’s OS version	CPU	CPU Family	CPU Micro-architecture	Analysis
Dell Latitude 5400	Archon OS v3.0.0.2	Intel® Core™ i5-8365U	Whiskey Lake	Skylake	Equivalent TOE models based on identical CPU microarchitecture: Skylake
Dell Latitude 5410	Archon OS v3.0.0.2	Intel® Core™ i7-10810U	Comet Lake	Skylake	
Dell Latitude 5430	Archon OS v3.0.0.2	Intel® Core™ i7-1255U	Alder Lake	Golden Cove	Equivalent TOE models based on equivalent CPU microarchitecture.
Dell Precision 3260	Archon OS v3.0.0.2	Intel® Core™ i7-12700	Alder Lake	Golden Cove	
Dell Precision 3570	Archon OS v3.0.0.2	Intel® Core™ i7-1255U	Alder Lake	Golden Cove	
Dell Latitude 5440	Archon OS v3.0.0.2	Intel® Core™ i5-1335U	Raptor Lake	Raptor Cove	
Dell Latitude 5540	Archon OS v3.0.0.2	Intel® Core™ i5-1335U	Raptor Lake	Raptor Cove	
Dell Precision 3580	Archon OS v3.0.0.2	Intel® Core™ i5-1350P	Raptor Lake	Raptor Cove	

Based on the equivalency rationale listed above, testing was performed in full on the below TOE hardware models.

Table 5: TOE Tested Hardware Models

TOE’s Model	TOE’s OS version	Instructions Set Extensions	CPU
Dell Latitude 5410	Archon OS v3.0.0.2	Intel® SSE4.1, Intel® SSE4.2, Intel® AVX2	Intel® Core™ i7-10810U
Dell Latitude 5430	Archon OS v3.0.0.2	Intel® SSE4.1, Intel® SSE4.2, Intel® AVX2	Intel® Core™ i7-1255U

5 Test Bed Descriptions

5.1 Test Time and Location

All testing was carried out at the Acumen Security offices located at 2400 Research Blvd Suite #395, Rockville, MD 20850. Testing occurred from January 2024 through June 2024 on TOE's version 3.0.0.0.

Regression testing was performed due to mitigating applicable CVEs in the new build 3.0.0.2 provided on June, 2024. The following tests were performed during regression testing to ensure ample coverage of all testing requirements:

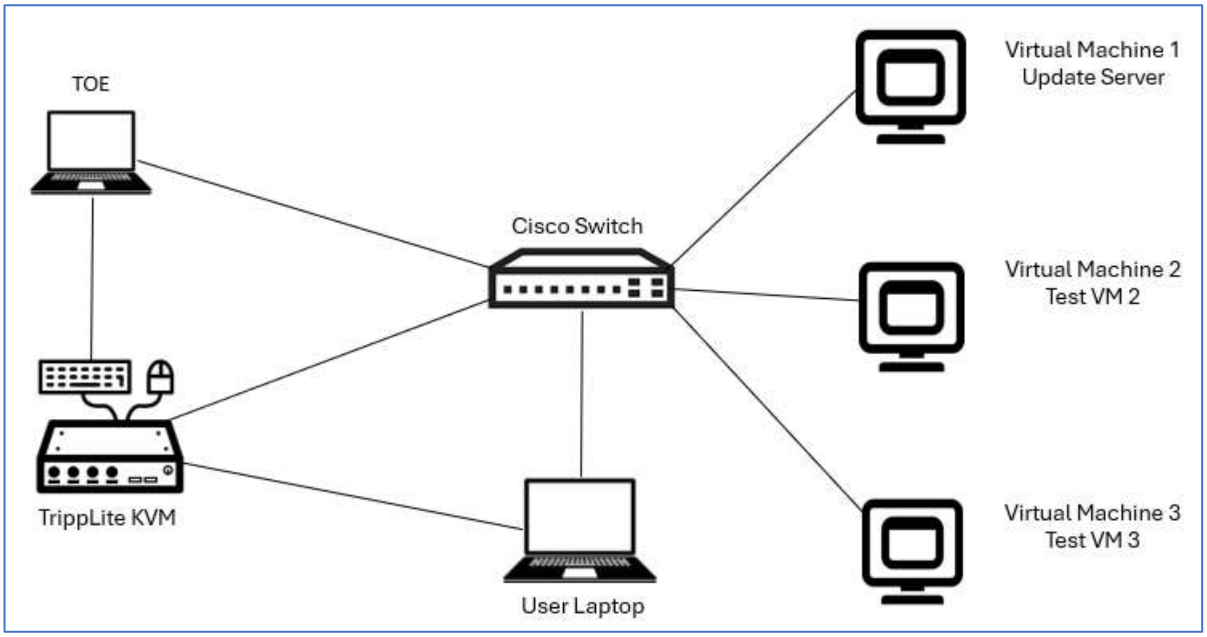
FPT_TST_EXT.1 Test#43
FPT_TUD_EXT.1 Test #47
FPT_TUD_EXT.1 Test #49
FCS_TLSC_EXT.2.1 Test#2
FIA_X509_EXT.1 Test#64

The 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 in a secure repository.

5.2 Test Bed

Below is a visual representation of the components included in the test bed:

Figure 2: Test Bed



5.3 Configuration Information

Table 6: Test Bed Configuration

Name	OS	Version	Function	Protocols	Tools
TOE	Archon OS	V3.0.0.2	TOE	Console	OpenSSL 1.1.1k FIPS 25 Mar 2021. VIM - Vi IMproved 8.0 annocheck: Version 11.13. gdb-8.2-19.el8.x86_64
Virtual Machine 1	Ubuntu Linux (64-bit)	Ubuntu 20.04.6 LTS Linux Kernel - 5.15.0-107-generic	Update Server	HTTPS, SSH, TLS	tcpdump version 4.9.3 OpenSSL 1.1.1f 31 Mar 2020 libpcap version 1.9.1 xxd V1.10 27oct98 Apache/2.4.41 (Ubuntu) acumen-tlsc 10/12/2021 acumen-tls version 3.0.0 x509-mod v1.1
Virtual Machine 2	Ubuntu Linux (64-bit)	Ubuntu 20.04.6 LTS Linux Kernel - 5.15.0-107-generic	Test VM 2	SSH, TLS	OpenSSL v1.1.1f, tcpdump v4.9.3
Virtual Machine 3	Ubuntu Linux (64-bit)	Ubuntu 22.04.3 LTS Linux Kernel - 5.15.0-101-generic	Test VM 3	TLS, SSH	libpcap version 1.10.1 OpenSSL 3.0.2 tcpdump version 4.99.1 Apache/2.4.52 (Ubuntu) x509-mod v1.1 acumen-tlsc, 10/12/2021 acumen-tls version 3.0.0
Cisco Switch	Cisco Catalyst 2960-L	ios 15.2	Gateway (Also acts like a router)	NA	NA
TrippLite KVM	NetCommander 16-Port Cat5 KVM over IP Switch	Firmware version 2.2.1263.1.0	Console Access to TOE	RS-232 connection to the TOE on Port 1 KVM with remote IP access	NA
User Laptop (HP Pavilion)	Windows	Windows 10 pro	Testing Laptop	SSH	MobaXterm V21.3 Wireshark Version 4.0.2 WinSCP V5.21.6

Name	OS	Version	Function	Protocols	Tools
					HxD Hex Editor Version 2.5.0.0 (x86-64) XCAv2.4.0

6 Detailed Test Cases (TSS and AGD Activities)

6.1 Mandatory Requirements

6.1.1 Audit Data Generation (FAU)

6.1.1.1 **FAU_GEN.1 Audit Data Generation (Refined)**

6.1.1.1.1 FAU_GEN.1 TSS

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

6.1.1.1.2 FAU_GEN.1 Guidance

Objective:

- The evaluator will check the administrative guide and ensure that it lists all of the auditable events. The evaluator will check to make sure that every audit event type selected in the ST is included.
- The evaluator will check the administrative guide and ensure that it provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator will ensure that the fields contain the information required.

Evaluator Findings:

- The evaluator examined the section titled **Audit Event Examples** in the AGD to verify that it lists all of the auditable events, including every audit event type selected in the ST.

  Upon investigation, the evaluator found that the AGD lists all audit events found in the ST.
- The evaluator examined the section titled **Audit Record Description** in the AGD to verify that it provides a format for audit records and that the fields contain the information required.

  Upon investigation, the evaluator found that the AGD provides an audit event format that includes keywords and a definition for each keyword.

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

Verdict:

PASS.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 **FCS_CKM.1 Cryptographic Key Generation (Refined)**

6.1.2.1.1 FCS_CKM.1 TSS (Applied TD0712)

Objective:

- The evaluator will ensure that the TSS identifies the key sizes supported by the OS.
- If the ST specifies more than one scheme, the evaluator will examine the TSS to verify that it identifies the usage for each scheme.
- If “P-256” is selected, the evaluator will examine the TSS to verify that it is only used for Bluetooth functions.

Evaluator Findings:

- The evaluator examined the section titled **TOE Summary Specification, FCS_CKM.1** in the Security Target to verify that the TSS identifies the key sizes supported by the OS.

  Upon investigation, the evaluator found that the TSS states that **The TOE implements RSA and ECC key generation and verification as specified in FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 and B.4 (respectively). The TOE implements FFC key generation as specified in NIST SP 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes". RSA key sizes of 3072 and 4096 are supported. ECC curve P-384 is supported. The FFC key size of L=3072 (Group 15) is supported.**

- The evaluator examined the section titled **TOE Summary Specification, FCS_CKM.1** in the Security Target to verify that it identifies the usage for each scheme.

  The TSS, FCS_CKM.1 refers to section 6.1, Table 13 of the ST that provides a mapping for each key and defines the type/usage and the identifies the source of the key and its intended usage.

- The evaluator examined the section titled **TOE Summary Specification, FCS_CKM.1** in the Security Target to verify that if "P-256" is selected then it is only used for Bluetooth functions.

  Upon investigation, the evaluator verified that 'P-256' was not selected.

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

Verdict:

PASS.

6.1.2.1.2 FCS_CKM.1 Guidance (Applied TD0712)

Objective:

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

Evaluator Findings:

- The evaluator examined the section titled **Configuring Archon OS into the CC Evaluated Configuration** in the AGD to verify that it instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all uses defined in this PP.

  Upon investigation, the evaluator found that the AGD specifies

  SCAP support and configuration combined with OSPP support and configuration means that by default, Archon OS v3.0.0.2 is configured with a subset of CC evaluated configuration parameters. Specifically, there are no TLS parameters that need to be configured (with the exception of certificates) and the TOE is automatically configured in FIPS mode.

  Specifically, the following is configured:

  - **the selected key generation schemes and key sizes,**
  - **the key establishment schemes,**
  - **the encryption/decryption modes and key sizes,**
  - **the supported TLS client cipher suites,**
  - **the supported groups extension, and**
  - **2048-bit RSA is used for secure boot signatures only.**

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

Verdict:

PASS.

6.1.2.2 **FCS_CKM.2 Cryptographic Key Establishment (Refined)**

6.1.2.2.1 FCS_CKM.2 TSS

Note, the following TSS activities are identified as part of the Test Activity for FCS_CKM.2.

Objective:

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

Evaluator Findings:

- The evaluator reviewed ST, sections 5.3.2.2 and 5.3.2.1 and determined that the supported key establishment schemes in FCS_CKM.2.1 correspond to the key generation schemes identified in FCS_CKM.1.1.
- The evaluator examined the section titled **TOE Summary Specification, FCS_CKM.2** in the ST to verify that the TSS identifies the usage for each scheme. Upon investigation, the evaluator found that **The TOE supports elliptic curve key establishment using the NIST curve P-384 during TLS mutual authentication when communicating with an update server or remote TLS servers.** Additionally, the TSS states **The TOE supports finite field key establishment using safe primes during TLS mutual authentication when communicating with update server or remote TLS servers.**

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

Verdict:

PASS.

6.1.2.2.2 FCS_CKM.2 Guidance

Note, the following Guidance activity is identified as part of Test Activity for FCS_CKM.2.

Objective:

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

Evaluator Findings:

- The evaluator examined the section titled **Configuring Archon OS into the CC Evaluated Configuration** in the AGD to verify that it instructs the administrator how to configure the OS to use the selected key establishment scheme(s).

  Upon investigation, the evaluator found that the AGD specifies **SCAP support and configuration combined with OSPP support and configuration means that by default, Archon OS v3.0.0.2 is configured with a subset of CC evaluated configuration parameters. Specifically, there are no TLS parameters that need to be configured (with the exception of certificates) and the TOE is automatically configured in FIPS mode.**

  Specifically, the following is configured:

  - **the selected key generation schemes and key sizes,**
  - **the key establishment schemes,**
  - **the encryption/decryption modes and key sizes,**
  - **the supported TLS client cipher suites,**
  - **the supported groups extension, and**

- **2048-bit RSA is used for secure boot signatures only.**

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

Verdict:

PASS.

6.1.2.3 FCS_CKM_EXT.4 Cryptographic Key Destruction

6.1.2.3.1 FCS_CKM_EXT.4 TSS (Applied TD0701)

Objective:

- The evaluator examines the TSS to ensure it describes how the keys are managed in volatile memory. This description includes details of how each identified key is introduced into volatile memory (e.g. by derivation from user input, or by unwrapping a wrapped key stored in nonvolatile memory) and how they are overwritten.
- The evaluator will check to ensure the TSS lists each type of key that is stored in non-volatile memory and identifies how the TOE interacts with the underlying platform to manage keys (e.g., store, retrieve, destroy). The description includes details on the method of how the TOE interacts with the platform, including an identification and description of the interfaces it uses to manage keys (e.g., file system APIs, platform key store APIs).
- If the ST makes use of the open assignment and fills in the type of pattern that is used, the evaluator examines the TSS to ensure it describes how that pattern is obtained and used. The evaluator will verify that the pattern does not contain any CSPs.
- The evaluator will check that the TSS identifies any configurations or circumstances that may not strictly conform to the key destruction requirement.
- If the selection “destruction of all key encrypting keys (KEKs) protecting the target key according to FCS_CKM_EXT.4.1, where none of the KEKs protecting the target key are derived" is included the evaluator will examine the TOE’s keychain in the TSS and identify each instance when a key is destroyed by this method. In each instance the evaluator will verify all keys capable of decrypting the target key are destroyed in accordance with a specified key destruction method in FCS_CKM_EXT.4.1. The evaluator will verify that all of the keys capable of decrypting the target key are not able to be derived to reestablish the keychain after their destruction.

Evaluator Findings:

- The evaluator examined section titled **TOE Summary Specification, FCS_CKM_EXT.4** in the Security Target to verify that the ST, section 6.1 and section 6 describe how the keys are managed in volatile memory, including the details of how each identified key is introduced into volatile memory and how they are overwritten.

  Upon investigation, the evaluator found that the TSS, **FCS_CKM_EXT.4**, states that: **For volatile memory, the TOE destroys keys and key material by performing a single overwrite consisting of zeroes.**

- The evaluator examined section titled **TOE Summary Specification, FCS_CKM_EXT.4** in the Security Target to verify that the TSS lists each type of key that is stored in non-volatile memory and identifies how the TOE interacts with the underlying platform to manage keys, including details on the method of how the TOE interacts with the platform, including an identification and description of the interfaces it uses to manage keys.

  Upon investigation, the evaluator found that the TSS states that: **For non-volatile memory, the TOE destroys keys and key material by performing an administrator configurable number (default 3) of**

overwrites of the logical storage location with a pseudo random pattern. The pseudo random pattern is generated by an ISAAC PRNG which is initialized from /dev/urandom. The TSS references Section 6.1, which contains Table 13 listing the types of keys/CSPs stored in non-volatile memory in the fourth column.

- The evaluator determined that there was no usage of an open assignment in the SFR FCS_CKM_EXT.4 to define a type of pattern.
- The evaluator examined the section titled **TOE Summary Specification, FCS_CKM_EXT.4** in the Security Target to verify that the TSS identifies any configurations or circumstances that may not strictly conform to the key destruction requirement.

  Upon investigation, the evaluator found that the TSS, **FCS_CKM_EXT.4** states **All instances of keys in non-volatile storage might not be deleted if the physical drive has replaced a sector containing a key with a spare sector. To minimize this risk, the physical drive should be end-of-life before a significant amount of damage to the drive's health can occur**.
- The evaluator examined the SFR FCS_CKM_EXT.4 and determined that the selection "destruction of all key encrypting keys (KEKs) protecting the target key according to FCS_CKM_EXT.4.1, where none of the KEKs protecting the target key are derived" was not included as a selection, therefore this assurance activity is considered not applicable.

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

Verdict:

PASS.

6.1.2.3.2 FCS_CKM_EXT.4 Guidance

Objective:

- There are a variety of concerns that may prevent or delay key destruction in some cases. The evaluator will check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS and any other relevant Required Supplementary Information.

  The evaluator will check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer and how such situations can be avoided or mitigated if possible.

  When the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-leveling and garbage collection. This may create additional copies of the key that are logically inaccessible but persist physically. In this case, to mitigate this the drive should support the TRIM command and implements garbage collection to destroy these persistent copies when not actively engaged in other tasks. Drive vendors implement garbage collection in a variety of different ways, as such there is a variable amount of time until data is truly removed from these solutions. There is a risk that data may persist for a longer amount of time if it is contained in a block with other data not ready for erasure. To reduce this risk, the operating system and file system of the OE should support TRIM, instructing the non-volatile memory to erase copies via garbage collection upon their deletion. If a RAID array is being used, only set-ups that support TRIM are utilized. If the drive is connected via PCI-Express, the operating system supports TRIM over that channel. The drive should be healthy and contains minimal corrupted data and should be end-of-lifed before a significant amount of damage to drive health occurs, this minimizes the risk that small amounts of potentially recoverable data may remain in damaged areas of the drive.

Evaluator Findings:

- The evaluator examined the section titled **Non-Volatile Drives and Keys** in the AGD to verify that it identifies configurations or circumstances that may not strictly conform to the key destruction requirement, that this description is consistent with the relevant parts of the TSS and any other relevant Required Supplementary Information, and that it provides guidance on situations where key destruction may be delayed at the physical layer and how such situations can be avoided or mitigated if possible.

  Upon investigation, the evaluator found that the Guidance states that **All instances of keys in non-volatile storage might not be deleted if the physical drive has replaced a sector containing a key with a spare sector. To minimize this risk, the physical drive should be end-of-life before a significant amount of damage to the drive's health can occur.**

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

Verdict:

PASS.

6.1.2.4 FCS_COP.1/ENCRYPT Cryptographic Operation - Encryption/Decryption (Refined)

6.1.2.4.1 FCS_COP.1/ENCRYPT TSS (Applied TD0712)

Objective:

- If "128-bit" is selected, the evaluator will examine the TSS to verify that 128-bit is only used with AES-CCM for Bluetooth functions.

Evaluator Findings:

- The evaluator examined the Security Target to verify that the TOE implements AES as specified in FIPS 197 with 256-bit key sizes. And 128 bit is not used for Bluetooth functions.

  Upon investigation, the evaluator found that the option "128-bit" was not selected in SFR FCS_COP.1/ENCRYPT in ST and the FCS_COP.1/ENCRYPT TSS section does not include a 128 bit claim, hence this assurance activity is considered not applicable.

Verdict:

PASS.

6.1.2.4.2 FCS_COP.1/ENCRYPT Guidance

Objective:

- The evaluator will verify that the AGD documents contain instructions required to configure the OS to use the required modes and key sizes.

Evaluator Findings:

- The evaluator examined the sections titled **Configuring Archon OS into the CC Evaluated Configuration** in the AGD to check if it contains instructions required to configure the OS to use the required modes and key sizes.

  Upon investigation, the evaluator found that **SCAP support and configuration combined with OSPP support and configuration means that by default, Archon OS v3.0.0.2 is configured with a subset of CC evaluated configuration parameters. Specifically, there are no TLS parameters that need to be configured (with the exception of certificates) and the TOE is automatically configured in FIPS mode.**

  Specifically, the following is configured:

  - **the selected key generation schemes and key sizes,**

- **the key establishment schemes,**
- **the encryption/decryption modes and key sizes,**
- **the supported TLS client cipher suites,**
- **the supported groups extension, and**
- **2048-bit RSA is used for secure boot signatures only.**

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

Verdict:

PASS.

6.1.2.5 **FCS_COP.1/HASH Cryptographic Operation - Hashing (Refined)**

6.1.2.5.1 FCS_COP.1/HASH TSS

Note, the following TSS activity is identified as part of the Test Activity for FCS_COP.1/HASH.

Objective:

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

Evaluator Findings:

- The evaluator examined the section titled **TOE Summary Specifications, FCS_COP.1/HASH** and section titled **Security Functional Requirements,** subsection **Cryptographic Support (FCS)** in the Security Target to verify that the TSS documents the association of the hash function with other application cryptographic functions.

  Upon investigation, the evaluator found that in TSS section **FCS_COP.1/HASH** the hashing algorithms are associated with other cryptographic functions and includes **SHA-384 is used to verify the integrity of TOE updates and is used in TLS key establishment and key agreement. SHA-512 is used for the kernel DRBG, boot integrity (Secure Boot), and user password protection.** Additionally, the TSS states **SHA-256 is supported in the TOE in order to be compatible with remote systems (e.g. TLS servers, CAs) using RSA certificates that specify the use of SHA-256. RSA certificates are supported, and they specify a SHA value used with signatures associated with them. RSA certificates can be imported into the system as trusted CA certs for cert chains, and they can be received dynamically from TLS servers during connection setup.**

  The evaluator found that the TSS section **FCS_COP.1/SIGN** states hashing algorithms are associated with digital signatures. The hashing algorithms identified in FCS_COP.1/SIGN are consistent with those used for SFR FCS_COP.1/HASH.

  In section SFR **FCS_COP.1/KEYHMAC**, the hashing algorithms are mapped to where they are used in HMACs. The hashing algorithms in SFR FCS_COP.1/KEYHMAC are consistent with those used for SFR FCS_COP.1/HASH.

  The evaluator found that the TSS section **FPT_TUD_EXT.x** describes trusted updates and their use of digital signatures and hashing. The SHA-384 hash algorithm in the TSS FPT_TUD_EXT.x section is consistent with the hashing algorithms in section FCS_COP.1/HASH.

  In section TSS **FCS_TLS_EXT.1, FCS_TLSC_EXT.x**, the TLS cipher suites are listed that include hashing algorithms (SHA-384) used for TLS communication. The evaluator considered each of the algorithms described in sections TSS FCS_TLS_EXT.1, FCS_TLSC_EXT.x and found they were consistent with the hashing algorithms claimed in FCS_COP.1/HASH.

  The evaluator found that the TSS section **FPT_TST_EXT.1** describes boot integrity and it’s use of digital signatures and hashing functions. The SHA-512 hash algorithm in the TSS FPT_TST_EXT.1 section is consistent with the hashing algorithms in section FCS_COP.1/HASH.

In section TSS **FCS_RBG_EXT.1/KERN** of the ST, the TOE makes use of an HMAC_DRBG which requires a HMAC function. The hashing function is identified as SHA-512 and is consistent with the claims made in SFR FCS_COP.1/HASH.

The evaluator found that the TSS section **FIA_UAU.5** passwords are hashed and used to validate user logins. The SHA-512 hash algorithm in the TSS FIA_UAU.5 section is consistent with the hashing algorithms in section FCS_COP.1/HASH.

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

Verdict:

PASS.

6.1.2.5.2 FCS_COP.1/HASH Guidance

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

6.1.2.6 **FCS_COP.1/KEYHMAC Cryptographic Operation - Keyed-Hash Message Authentication (Refined)**

6.1.2.6.1 FCS_COP.1.1/KEYHMAC TSS

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

6.1.2.6.2 FCS_COP.1.1/KEYHMAC Guidance

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

6.1.2.7 **FCS_COP.1/SIGN Cryptographic Operation - Signing (Refined)**

6.1.2.7.1 FCS_COP.1.1/SIGN TSS (Applied TD0809)

Objective:

- [Conditional: if "2048-bit (for secure boot only) or greater" is selected]. The evaluator shall check that the TSS documents that 2048-bit RSA is used only for secure boot and a greater key size is used for any other functions.

Evaluator Findings:

- The evaluator examined the section titled **TOE Summary Specifications, FCS_COP.1/SIGN** in the Security Target to verify that the TSS documents that 2048-bit RSA is used only for secure boot and a greater key size is used for any other functions.

  Upon investigation, the evaluator found that the TSS states **RSA key sizes of 2048 (for secure boot only), 3072, and 4096 are supported, utilizing SHA-256, SHA-384, and SHA-512 hashing algorithms**.

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

Verdict

PASS.

6.1.2.7.2 FCS_COP.1.1/SIGN Guidance (TD0809 Applied)

Objective:

- [Conditional: if "2048-bit (for secure boot only) or greater" is selected] The evaluator shall check that the AGD documents any configuration needed to ensure 2048-bit RSA is used only for secure boot and a greater key size is used for any other functions.

Evaluator Findings:

- The evaluator examined the section titled **Secure Boot** in the AGD to verify that it documents any configuration needed to ensure 2048-bit RSA is used only for secure boot and a greater key size is used for any other functions.

  Upon investigation, the evaluator found that the AGD states that: **The second portion of the boot software is signed by a central Certificate Authority using a 4096-bit SHA-512 signature.**

  The evaluator examined the section titled **Secure Boot** in the AGD to verify that it documents any configuration needed to ensure 2048-bit RSA is used only for secure boot and a greater key size is used for any other functions.

  Upon investigation, the evaluator found that the AGD states that: **RSA-2048 is only supported for secure boot; the certs related to secure boot are preloaded by Dell. No configuration by the administrator is necessary.**

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

Verdict:

PASS.

6.1.2.8 FCS_RBG_EXT.1/OSSL Random Bit Generation

6.1.2.8.1 FCS_RBG_EXT.1/OSSL TSS

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

6.1.2.8.2 FCS_RBG_EXT.1/OSSL Guidance

Note, the following Guidance activity is identified as part of the Test Activity for FCS_RBG_EXT.1.

Objective:

- The evaluator will also confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality.

Evaluator Findings:

- The evaluator examined the section titled **Configuring Archon OS into the CC Evaluated Configuration** in the AGD to confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality.

  Upon investigation, the evaluator found that the AGD states that **The TOE is automatically configured in FIPS mode which ensures the system generates all keys (RNG functionality) using FIPS approved algorithms. No other configuration is required to configure RNG functionality**.

6.1.2.9 FCS_RBG_EXT.1/KERN Random Bit Generation

6.1.2.9.1 FCS_RBG_EXT.1/KERN TSS

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

6.1.2.9.2 FCS_RBG_EXT.1/KERN Guidance

Note, the following Guidance activity is identified as part of the Test Activity for FCS_RBG_EXT.1.

Objective:

- The evaluator will also confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality.

Evaluator Findings:

- The evaluator examined the section titled **Configuring Archon OS into the CC Evaluated Configuration** in the AGD to confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality.

  Upon investigation, the evaluator found that the AGD states that **The TOE is automatically configured in FIPS mode which ensures the system generates all keys (RNG functionality) using FIPS approved algorithms. No other configuration is required to configure RNG functionality**.

6.1.2.10 **FCS_STO_EXT.1 Storage of Sensitive Data**

6.1.2.10.1 FCS_STO_EXT.1 TSS

Objective:

- The evaluator will check the TSS to ensure that it lists all persistent sensitive data for which the OS provides a storage capability. For each of these items, the evaluator will confirm that the TSS lists for what purpose it can be used, and how it is stored. The evaluator will confirm that cryptographic operations used to protect the data occur as specified in FCS_COP.1/ENCRYPT.

Evaluator Findings:

- The evaluator examined the section titled **TOE Summary Specifications, FCS_STO_EXT.1** in the Security Target to verify that the TSS lists all persistent sensitive data for which the OS provides a storage capability.

  Upon investigation, the evaluator found that the TSS states that **The TOE includes the OpenSSL library to securely store sensitive data. OpenSSL provides file encryption services using AES-256 in CBC mode. Sensitive data is passwords and keys and can be found in the /etc directory which contains system-wide configuration files and system databases. Access to the files in /etc is limited with strict file permissions and/or encryption.**

  The evaluator found the TSS states **Passwords are used for local user authentication. Keys are used in TLS key agreement and key establishment and signature verification**.

  The evaluator verified, through testing, that the local files are protected.

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

Verdict:

PASS.

6.1.2.10.2 FCS_STO_EXT.1 Guidance

Objective:

- The evaluator will consult the developer documentation to verify that instructions exist on applications that should securely store credentials.

Evaluator Findings:

- The evaluator examined the section titled **Storage of Sensitive Data** in the AGD to verify that instructions exist on applications that should securely store credentials.

Upon investigation, the evaluator found that the AGD states that **Archon OS follows standard conventions for storing sensitive data. Applications must store their sensitive data in the /etc directory with restrictive access permissions. Access to sensitive data should be restricted to root and/or the application storing the sensitive data. Sensitive data is keys and passwords.**

In addition, in section **Called by CLI**, the evaluator found that **Archon OS also provides the ability to encrypt/decrypt sensitive files using OpenSSL.**

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

Verdict:

PASS.

6.1.2.11 **FCS_TLS_EXT.1 TLS Protocol**

6.1.2.11.1 FCS_TLS_EXT.1.1 TSS

According to the Functional Package, there are no TSS AA requirements for this SFR.

6.1.2.11.2 FCS_TLS_EXT.1.1 Guidance

Objective:

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

Evaluator Findings:

- The evaluator examined the section titled **Cryptographic Library Configuration** and ensured that the selections indicated in the ST are consistent with selections in the dependent components.

  Upon investigation, the evaluator found that the AGD states that **The TOE acts as a TLS Client communicating with an Update Server and remote servers**.

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

Verdict:

PASS.

6.1.3 User Data Protection (FDP)

6.1.3.1 **FDP_ACF_EXT.1 Access Controls for Protecting User Data**

6.1.3.1.1 FDP_ACF_EXT.1.1 TSS

Objective:

- The evaluator will confirm that the TSS comprehensively describes the access control policy enforced by the OS. The description must include the rules by which access to particular files and directories are determined for particular users.
- The evaluator will inspect the TSS to ensure that it describes the access control rules in such detail that given any possible scenario between a user and a file governed by the OS the access control decision is unambiguous.

Evaluator Findings:

- The evaluator examined section titled **TOE Summary Specification, FDP_ACF_EXT.1** in the Security Target to verify that the TSS comprehensively describes the access control policy enforced by the OS, including the rules by which accesses to files and directories are determined for particular users.

  Upon investigation, the evaluator found that the TSS, section FDP_ACF_EXT.1, thoroughly describes the access policy as standard UNIX permission bits, defining access for read, write, and execute permissions, with automatic blocking of write access to filesystems mounted as read-only.

- The evaluator also examined section titled **TOE Summary Specification, section FDP_ACF_EXT.1** in the Security Target to verify that the TSS describes the access control rules in such detail that given any possible scenario between a user and a file governed by the OS the access control decision is unambiguous.

  Upon investigation, the evaluator found in the TSS includes descriptions about

  - the "umask" attribute used to determine the default access permission for new objects;
  - POSIX-type Access Control Lists used to define a fine-grained access control on a per-file or per-directory basis; and
  - the additional access control bits of "SUID", "SGID", and "SAVETXT" that are used by the kernel.

  The TSS also describes the files and filesystems to be protected. The evaluator verified that the description of the access control rules is sufficiently detailed and that all scenarios of access control from users to files are unambiguously identified.

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

Verdict:

PASS.

6.1.3.1.2 FDP_ACF_EXT.1.1 Guidance

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

6.1.4 Identification and Authentication (FIA)

6.1.4.1 **FIA_AFL.1 Authentication failure handling (Refined)**

6.1.4.1.1 FIA_AFL.1 TSS

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

6.1.4.1.2 FIA_AFL.1 Guidance

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

6.1.4.2 **FIA_UAU.5 Multiple Authentication Mechanisms (Refined)**

6.1.4.2.1 FIA_UAU.5 TSS

Note, the first bullet item of the TSS activity is identified as part of the Test Activities for FIA_UAU.5.

Objective:

- The evaluator will examine the TSS for guidance on supported protected storage and will then configure the TOE or OE to establish a PIN which enables release of the asymmetric key from the protected storage (such as a TPM, a hardware token, or isolated execution environment) with which the OS can interface.

- The evaluator will ensure that the TSS describes the rules as to how each authentication mechanism specified in FIA_UAU.5.1 is implemented and used. Example rules are how the authentication mechanism authenticates the user (i.e. how does the TSF verify that the correct password or authentication factor is used), the result of a successful authentication (i.e. is the user input used to derive or unlock a key) and which authentication mechanism can be used at which authentication factor interfaces (i.e. if there are times, for example, after a reboot, that only specific authentication mechanisms can be used). Rules regarding how the authentication factors interact in terms of unsuccessful authentication are covered in FIA_AFL.1.

Evaluator Findings:

- N/A, because the option "authentication based on username and a PIN that releases an asymmetric key" is not selected in FIA_UAU.5 SFR.
- The evaluator examined the section titled **TOE Summary Specification, FIA_UAU.5** in the Security Target to verify that the TSS describes each mechanism provided to support user authentication and the rules describing how the authentication mechanism(s) provide authentication, the result of a successful authentication and which authentication mechanism can be used at which authentication factor interfaces.

  Upon investigation, the evaluator found that the TSS states that **The TOE supports authentication based on username and password at the local console. The TOE performs username and password authentication using a local set of credentials. During password-based login, a PAM (Pluggable Authentication Module) module is invoked which collects the username and password. The pam_unix module verifies the user is located in the password database file /etc/passwd and compares a hash (SHA-512) of the provided password with one previously stored in the file /etc/shadow. If successful, a user session is started. Otherwise, a delay occurs before allowing another attempt if permitted.**

  With regards to the rules regarding how the authentication factors interact in terms of unsuccessful authentication are covered in FIA_AFL.1, the evaluator determined that **The TOE will detect when an administrator configurable integer (/etc/security/faillock.conf file deny parameter) within 1-65,535 unsuccessful authentication attempts for authentication based on username and password occur related to password-based authentication at the local console. Once the specified number of unsuccessful authentication attempts for an account has been met, the TOE locks the account.**

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

Verdict:

PASS.

6.1.4.2.2 FIA_UAU.5 Guidance

Objective:

- The evaluator will verify that configuration guidance for each authentication mechanism is addressed in the AGD guidance.

Evaluator Findings:

- The evaluator examined the sections titled **User/Administrator Accounts** and in the AGD to verify that it addresses configuration guidance for each authentication mechanism.

  Upon investigation, the evaluator found that the AGD section titled **User/Administrator Accounts** states that **The TOE only supports local logins using username and password at the local console. It does not support remote administration.** Additionally, the section titled **Creating/Deleting User Accounts** includes instructions about how to create a user account. **The administrator can create user accounts using the**

`useradd [options] <user_name>` command. The user account will be locked and password-less. Once a user account has been created, the administrator can make this account an administrator by adding it to the wheel group by running `usermod -aG wheel <username>`.

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

Verdict:

PASS.

6.1.4.3 FIA_X509_EXT.1 X.509 Certificate Validation

6.1.4.3.1 FIA_X509_EXT.1 TSS (Applied TD0773)

Objective:

- The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm.
- If there are exceptional use cases where the OS cannot perform revocation checking in accordance with at least one of the revocation methods, the evaluator shall ensure the TSS describes each revocation checking exception use case and, for each exception, the alternate functionality the TOE implements to determine the status of the certificate and disable functionality dependent on the validity of the certificate.

Evaluator Findings:

- The evaluator examined the section titled **TOE Summary Specifications, FIA_X509_EXT.1** in the Security Target to ensure that it describes where the check of validity of the certificates takes place. The evaluator ensured that the TSS also provides a description of the certificate path validation algorithm.

  Upon investigation, the evaluator found that the TSS states that **The certificate validity check is performed when the TOE receives the certificate during a TLS handshake.**

  The X.509 certificates are validated using the certificate path validation algorithm defined in RFC 5280 which can be summarized as follows:

  - **the public key algorithm and parameters are checked,**
  - **the current date/time is checked against the validity period,**
  - **revocation status is checked using CRL,**
  - **issuer name of X matches the subject name of X+1,**
  - **extensions are processed.**

- The evaluator examined the section titled **TOE Summary Specifications** in the Security Target to ensure that, if the OS cannot perform revocation in accordance with one of the revocation methods, the TSS describes each revocation checking exception use case, and for each exception, the alternate functionality the TOE implements to determine the status of the certificate and disable functionality dependent on the validity of the certificate.

  Upon investigation, the evaluator found that the TSS states that **If the validity check of a certificate fails, or if the TOE is unable to retrieve a valid and current CRL file from the CRL distribution point, the certificate is rejected. The TOE always verifies server certificates and always refuses to establish a trusted channel if the verification fails or if the TOE is unable to retrieve a valid and current CRL. There is not an override option**.

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

Verdict:

PASS.

6.1.4.3.2 FIA_X509_EXT.1 Guidance

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

6.1.4.4 **FIA_X509_EXT.2 X.509 Certificate Authentication**

6.1.4.4.1 FIA_X509_EXT.2 TSS

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

6.1.4.4.2 FIA_X509_EXT.2 Guidance

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

6.1.5 **Security Management (FMT)**

6.1.5.1 **FMT_MOF_EXT.1 Management of security functions behavior**

6.1.5.1.1 FMT_MOF_EXT.1 TSS

Objective:

- The evaluator will verify that the TSS describes those management functions that are restricted to Administrators, including how the user is prevented from performing those functions, or not able to use any interfaces that allow access to that function.

Evaluator Findings:

- The evaluator examined the section titled **TOE Summary Specifications, FMT_MOF_EXT.1** in the Security Target to verify that the TSS describes those management functions that are restricted to Administrators, including how the user is prevented from performing those functions, or not able to use any interfaces that allow access to that function.

  Upon investigation, the evaluator found that the TSS states that: **The TOE restricts all "Administrator" management activities listed in FMT_SMF_EXT.1 to users who are members of the "wheel" group. Members of this group are considered the administrators, because group membership allows users to elevate their privileges, allowing management of the TOE, using the sudo command.**

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

Verdict:

PASS.

6.1.5.1.2 FMT_MOF_EXT.1 Guidance

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

6.1.5.2 **FMT_SMF_EXT.1 Specification of Management Functions**

6.1.5.2.1 FMT_SMF_EXT.1 TSS

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

6.1.5.2.2 FMT_SMF_EXT.1 Guidance

Objective:

- The evaluator will verify that every management function captured in the ST is described in the operational guidance and that the description contains the information required to perform the management duties associated with the management function.

Evaluator Findings:

- The evaluator examined the sections in the AGD to verify that every management function identified in the ST is described and that the description contains the information required to perform the management duties associated with the management function. The following table identifies the section in the AGD that identifies the guidance.

Table 7 – Specification of Management Functions

Management Function	AGD Section
1. Enable/disable [session timeout]	Section **Enable Session Timeout** for enable and section **Disable Session Timeout** for disable**.**
2. Configure [session] inactivity timeout	Section **Inactivity Timeout**
3. Import keys/secrets into the secure key storage.	Section **Storing Certificates**
4. Configure local audit storage capacity	Section **Local Audit Storage Settings**
5. Configure minimum password length	Section **Configure Password Policy, minlen parameter.**
6. Configure minimum number of special characters in password	Section **Configure Password Policy, ocredit parameter.**
7. Configure minimum number of numeric characters in password	Section **Configure Password Policy, dcredit parameter.**
8. Configure minimum number of uppercase characters in password	Section **Configure Password Policy, ucredit parameter.**
9. Configure minimum number of lowercase characters in password	Section **Configure Password Policy, lcredit parameter.**
10. Configure lockout policy for unsuccessful authentication attempts through [timeouts between attempts]	Section **Failed Authentication Timeout**

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

Verdict:
PASS.

6.1.6 Protection of Security Functions (FPT)

6.1.6.1 FPT_ACF_EXT.1 Access controls

6.1.6.1.1 FPT_ACF_EXT.1.1 TSS

Objective:

- The evaluator will confirm that the TSS specifies the locations of kernel drivers/modules, security audit logs, shared libraries, system executables, and system configuration files. Every file does not need to be individually identified, but the system's conventions for storing and protecting such files must be specified.

Evaluator Findings:

- The evaluator examined the section titled **TOE Summary Specifications, FPT_ACF_EXT.1** in the Security Target to verify that the TSS specifies the locations of kernel drivers/modules, security audit logs, shared libraries, system executables, and system configuration files.

  Upon investigation, the evaluator found that the TSS states that **The TOE uses the file/directory permissions described in FDP_ACF_EXT.1 to prevent unprivileged users from modifying:**

  - **Kernel and its drivers/modules**
  - **Security audit logs**
  - **Shared libraries**
  - **System executables**
  - **System configuration files**

  The evaluator examined the TSS section for FDP_ACF_EXT.1 and verified the section describes the conventions used for storing and protecting the files identified above. Specifically, FDP_ACF_EXT.1 TSS section states: **The TOE uses these permissions to protect the following from unauthorized modification:**

  - **Kernel, drivers, and kernel modules – files in:**
    - ***/boot/***
    - ***/usr/lib/modules/***
    - ***/usr/lib/firmware/***
  - **Security audit logs – files in:**
    - ***/var/log/audit/***
    - ***/var/log/***
  - **Shared libraries – files in:**
    - ***/usr/lib64/***
    - ***/usr/lib/***
  - **System executables – files in:**
    - ***/usr/sbin/***
    - ***/usr/bin/***
    - ***/usr/libexec/***
  - **System configuration files – files in:**
    - ***/etc/***
    - ***/usr/lib/***

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

Verdict:

PASS.

6.1.6.1.2 FPT_ACF_EXT.1.1 Guidance

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

6.1.6.2 **FPT_ASLR_EXT.1 Address Space Layout Randomization**

6.1.6.2.1 FPT_ASLR_EXT.1 TSS

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

6.1.6.2.2 FPT_ASLR_EXT.1 Guidance

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

6.1.6.3 **FPT_SBOP_EXT.1 Stack Buffer Overflow Protection**

6.1.6.3.1 FPT_SBOP_EXT.1 TSS

Note, the following two TSS activities are identified as part of the Test Activity for FPT_SBOP_EXT.1.

Objective:

- For stack-based OSes, the evaluator will determine that the TSS contains a description of stack-based buffer overflow protections used by the OS. These are referred to by a variety of terms, such as stack cookie, stack guard, and stack canaries. The TSS must include a rationale for any binaries that are not protected in this manner
- For OSes that store parameters/variables separately from control flow values, the evaluator will verify that the TSS describes what data structures control values, parameters, and variables are stored. The evaluator will also ensure that the TSS includes a description of the safeguards that ensure parameters and variables do not intermix with control flow values.

Evaluator Findings:

- The evaluator examined the section titled **TOE Summary Specifications, FPT_SBOP_EXT.1** in the Security Target to verify that the TSS contains a description of stack-based buffer overflow protections used by the TOE and includes a rationale for any binaries that are not protected in this manner.

  Upon investigation, the evaluator found that the TSS states that **The TOE is a stack-based OS and is compiled with the option "stack-protector-strong" to add a stack canary and associated verification code during the entry and exit of function frames to prevent stack-based buffer overflows.**

  Also, on the same section the evaluator found that the TSS states that **The ST section 6.3 also lists all binaries not protected by stack mashing protections in use by the TOE, and their rationales for exclusion.**
- The second objective listed above does not apply because the **TOE Summary Specifications** specifies that **The TOE does not store parameters/variables separately from control flow values** and therefore, this assurance activity is not applicable.

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

Verdict:

PASS.

6.1.6.3.2 FPT_SBOP_EXT.1 Guidance

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

6.1.6.4 **FPT_TST_EXT.1 Boot Integrity**

6.1.6.4.1 FPT_TST_EXT.1 TSS

Objective:

- The evaluator will verify that the TSS section of the ST includes a comprehensive description of the boot procedures, including a description of the entire bootchain, for the TSF.

- The evaluator will ensure that the OS cryptographically verifies each piece of software it loads in the bootchain to include bootloaders and the kernel. Software loaded for execution directly by the platform (e.g. first-stage bootloaders) is out of scope. For each additional category of executable code verified before execution, the evaluator will verify that the description in the TSS describes how that software is cryptographically verified.
- The evaluator will verify that the TSS contains a description of the protection afforded to the mechanism performing the cryptographic verification.

Evaluator Findings:

- The evaluator examined the section titled **TOE Summary Specifications, FPT_TST_EXT.1** in the Security Target to verify that the TSS includes a comprehensive description of the boot procedures, including a description of the entire bootchain, for the TSF.

  Upon investigation, the evaluator found that the TSS includes a comprehensive description of the boot procedures, including a description of the entire bootchain, for the TSF.

- The evaluator examined the section titled **TOE Summary Specifications, FPT_TST_EXT.1** in the Security Target to verify that the TSS states that the OS cryptographically verifies each piece of software it loads in the bootchain to include bootloaders and the kernel. For each additional category of executable code verified before execution, the evaluator verifies that the description in the TSS describes how that software is cryptographically verified.

  Upon investigation, the evaluator found that the TSS states that **The boot chain consists of the following steps:**

  - **Hardware responsibility**
    - **Firmware initialization**
    - **First stage boot loader (shim.efi)**
  - **TOE responsibility**
    - **Second Stage Boot Loader (GRUB 2)**
    - **First root filesystem (initramfs)**
    - **Linux kernel (drivers and modules)**

  Secure Boot is a UEFI firmware security feature developed by the UEFI Consortium that ensures only immutable and signed software is loaded during the boot time.
  The first application loaded by the platform's firmware is the signed and trusted first-stage boot loader (shim.efi). This shim package itself holds the signing certificate and its own databases of trusted keys and hashes that are allowed to be loaded.
  The shim package's signature is verified by the signing certificate's RSA 2048 public key included in the shim package. The shim then uses this public key to verify the signature on the code signing public key held in the database. This code signing key is used to verify the signature of the second-stage boot loader, GRUB 2 (grubx64.efi).
  Next, GRUB 2 uses the code signing key to verify the signature on the first root filesystem (initramfs). Initramfs then uses RSA 4096 code (SHA 512) signing keys, from the database, to verify the signatures of the OS kernel.

- The evaluator examined the section titled **TOE Summary Specifications, FPT_TST_EXT.1** in the Security Target to verify that the TSS contains a description of the protection afforded to the mechanism performing the cryptographic verification.

  The evaluator examined the section and determined the first key in the key chain is obtained from the first application loaded by the platform and is considered trusted.

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

Verdict:

PASS.

6.1.6.4.2 FPT_TST_EXT.1 Guidance

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

6.1.6.5 **FPT_TUD_EXT.1 Trusted Update**

6.1.6.5.1 FPT_TUD_EXT.1 TSS

According to the PP, the TSS Evaluation Activities is part of the Testing assurance activity for FPT_TUD_EXT.1.

6.1.6.5.2 FPT_TUD_EXT.1 Guidance

According to the PP, there are no Guidance Evaluation Activities required for this FPT_TUD_EXT.1.

6.1.6.6 **FPT_TUD_EXT.2 Trusted Update for Application Software**

6.1.6.6.1 FPT_TUD_EXT.2 TSS

Objective:

- All origins supported by the OS must be indicated in the TSS and evaluated. However, this only includes those mechanisms for which the OS is providing a trusted installation and update functionality. It does not include user or administrator-driven download and installation of arbitrary files.

Evaluator Findings:

- The evaluator examined the section titled **TOE Summary Specifications, FPT_TUD_EXT.2** in the Security Target to verify that the TSS indicates all supported origins for updates.

  Upon investigation, the evaluator found that the TSS states that **The TOE has the ability to check for updates to itself. Updates are verified by RSA 4096 with SHA-384 prior to installation. Updates to the TOE and application software are downloaded by the TOE from the Update Server.**

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

Verdict:

PASS.

6.1.6.6.2 FPT_TUD_EXT.2 Guidance

Objective:

- The evaluator will check for updates to application software using procedures described in the documentation and verify that the OS provides a list of available updates. Testing this capability may require temporarily placing the system into a configuration in conflict with secure configuration guidance which specifies automatic update.

Evaluator Findings:

- The evaluator examined the section titled **System Updates** in the AGD to verify that it describes procedures to check for an update to application software.

  Upon investigation, the evaluator found that AGD section **Checking for Available TOE Updates** describes how to check for TOE updates.

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

Verdict:
PASS.

6.1.7 Trusted Path/Channels (FTP)

6.1.7.1 FTP_ITC_EXT.1 Trusted channel Communication

6.1.7.1.1 FTP_ITC_EXT.1 TSS

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

6.1.7.1.2 FTP_ITC_EXT.1 Guidance

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

6.1.7.2 FTP_TRP.1 Trusted Path

6.1.7.2.1 FTP_TRP.1 TSS (Applied TD0839)

Objective:

- The evaluator will examine the TSS to determine that the methods of remote or local OS administration are indicated, along with how those communications are protected.
- (Conditional: if "remote" is selected in FTP_TRP.1.1) The evaluator will also confirm that all protocols listed in the TSS in support of OS administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

Evaluation Findings:

- The evaluator examined the section titled **TOE Summary Specifications, FTP_TRP.1** in the Security Target to verify that the TSS indicates the methods of remote OS administration, along with how those communications are protected.

  Upon investigation, the evaluator found that the TSS, section **FTP_TRP.1** states that **The TOE provides a trusted path for local users. The TOE only supports local (keyboard) access which is considered a trusted interface.**
- N/A, because "remote" is not selected in the FTP_TRP.1 SFR**.**

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

Verdict:

PASS.

6.1.7.2.2 FTP_TRP.1 Guidance (Applied TD0839)

Objective:

- The evaluator will confirm that the operational guidance contains instructions for establishing remote administrative sessions or initial user authentication for each supported method.

Evaluator Findings:

- The evaluator examined the section titled **User/Administrator Accounts** in the AGD to verify that it contains instructions for establishing the remote administrative sessions or initial user authentication for each supported method.

  Upon investigation, the evaluator found that the AGD activity for **User/Administrator Accounts** states that **The TOE only supports local logins using username and password at the local console. It does not support remote administration.** Additionally, the section includes instructions in section **Creating/Deleting User Account,** about how to create a user account. **The administrator can create user accounts using the "useradd [options]** ***user_name*****" command. The user account will be locked and password-less. Once a user account has been created, the administrator can make this account an administrator by adding it to the wheel group by running "usermod -aG wheel <username>".**

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

Verdict:
PASS.

Strictly Optional Requirements

6.1.8 TOE Access (FTA)

6.1.8.1 FTA_TAB.1 Default TOE access banners

6.1.8.1.1 FTA_TAB.1 TSS

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

6.1.8.1.2 FTA_TAB.1 Guidance

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

6.2 Objective Optional Requirements

6.2.1 Protection of the TSF (FPT)

6.2.1.1 FPT_SRP_EXT.1 Software Restriction Policies

6.2.1.1.1 FPT_ SRP_EXT.1.1 TSS

Objective:

- The evaluator will ensure that the description of the supported characteristics in the TSS is consistent with the SFR.
- The evaluator will also ensure that any characteristics specified by the ST-author are described in sufficient detail to understand how to test those characteristics.

Evaluation Findings:

- The evaluator examined the section titled **TOE Summary Specifications, FPT_SRP_EXT.1** to verify that the description of the supported characteristics in the TSS matches with that of the SFR.

Upon investigation, the evaluator found that the TSS states that **The TOE includes a daemon, fapolicyd, that determines access rights to files based on a trust database and file or process attributes. By default, all applications that are packaged by rpm are automatically trusted. The user guidance provides instructions that enable the administrator to configure fapolicyd. The administrator is instructed to configure fapolicyd at TOE installation. Once configured, fapolicyd will create a file, compiled.rules, that identifies the trust/untrusted status of the files.** This is consistent with the section FPT_SRP_EXT.1 Software Restriction Policies of the ST which mentions file path as the only supported method.

- The evaluator examined the section titled **TOE Summary Specifications, FPT_SRP_EXT.1** to verify that any characteristics specified by the ST-author are described in sufficient detail to understand how to test those characteristics.

  Upon investigation, the evaluator found that the TSS states that **The user guidance provides instructions that enable the administrator to configure `fapolicyd`. The administrator is instructed to configure `fapolicyd` at TOE installation. Once configured, `fapolicyd` will create a file, `compiled.rules`, that identifies the trust/untrusted status of the files.**

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

Verdict:

PASS.

6.2.1.1.2 FPT_ SRP_EXT.1.1 Guidance

Objective:

- The evaluator will ensure that that the characteristics are described in sufficient detail for administrators to configure policies using them, and that the list of characteristics in the guidance is consistent with the information in the TSS.

Evaluator Findings:

- The evaluator examined the section titled **Software Restriction Policies (fapolicyd**) in the AGD to verify if the characteristics are described in sufficient detail for administrators to configure policies using them, and that the list of characteristics in the guidance is consistent with the information in the TSS.

  Upon investigation, the evaluator found that the AGD states that the **Fapolicyd is a daemon that determines whether or not access to files or execution of programs is allowed based on the software's reputation**. **By default, all applications that are packaged by rpm are automatically trusted and therefore, the following steps must be followed to enable fapolicyd policy checks.**

  The AGD includes the description of the keywords and syntax that the administrator must configure in order to configure fapolicyd.

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

Verdict:

PASS.

6.3 Selection-Based Requirements

6.3.1 Cryptographic Support (FCS)

6.3.1.1 FCS_TLSC_EXT.1 TLS Client Protocol

6.3.1.1.1 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 section titled **TOE Summary Specifications, FCS_TLSC_EXT.1** in the ST to ensure that the cipher suites supported are specified and that the cipher suites specified include those listed for this component.

  Upon investigation, the evaluator found that the TSS states that: **The TOE provides a TLSv1.2 client implementation with the following ciphersuites: TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288, TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289.** The evaluator confirmed that these ciphersuites are consistent with the allowable set of ciphersuites permitted in the SFR.

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

Verdict:

PASS.

6.3.1.1.2 FCS_TLSC_EXT.1.1 Guidance

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 section titled **Configuring Archon OS into the CC Evaluated Configuration** in the AGD and ensured that it contains instructions on configuring the product so that TLS conforms to the description in the TSS.

  Upon investigation, the evaluator found that **SCAP support and configuration combined with OSPP support and configuration means that by default, Archon OS v3.0.0.2 is configured with a subset of CC evaluated configuration parameters. Specifically, there are no TLS parameters that need to be configured (with the exception of certificates) and the TOE is automatically configured in FIPS mode.**

  Specifically, the following is configured:

  - **the selected key generation schemes and key sizes,**
  - **the key establishment schemes,**
  - **the encryption/decryption modes and key sizes,**
  - **the supported TLS client cipher suites,**
  - **the supported groups extension, and**
  - **2048-bit RSA is used for secure boot signatures only.**

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

Verdict:

PASS.

6.3.1.1.3 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 section titled **TOE Summary Specifications, FCS_TLSC_EXT.1** in the ST 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.

  Upon investigation, the evaluator found that the TSS states that: **The TOE establishes the reference identifier by parsing the DNS Name or IP address for the configured TLS server. The reference identifier is matched against the SAN, if present. If the SAN is not present, the referenced identifier is matched against the CN for DNS. For IP address, the TOE matches the identifier against the SAN only. The TOE supports wildcards in the DNS name of the server certificate (the left-most component in the presented certificate may be a wildcard (i.e. "*")).**
- The evaluator reviewed the TSS section titled **TOE Summary Specifications, FCS_TLSC_EXT.1** in the ST to ensure that it identifies whether and the manner in which certificate pinning is supported or used by the product.

  Upon investigation, the evaluator found that the TSS states that **The TOE does not support URI reference identifiers, SRV reference identifiers, or certificate pinning.**

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

Verdict:

PASS.

6.3.1.1.4 FCS_TLSC_EXT.1.2 Guidance

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 examined the section titled **User Initiated TLS Sessions** in the AGD and ensured that it includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS.

  Upon investigation, the evaluator found that the AGD activity mentions the OpenSSL command where **`--verify_hostname [hostname]`** configures the hostname that the TOE will convert into a DNS-ID and CN reference identifier. The left-most component in the presented certificate may be a wildcard (i.e. "*") and the option **`--verify_ip [IP address]`** configures the IP address that the TOE will convert into an IP address SAN reference identifier.

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

Verdict:

PASS.

6.3.1.1.5 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 selection of authorizing override of invalid certificates is not made in the FCS_TLSC_EXT.1.3 SFR.

Based on this finding, this assurance activity is considered not applicable.

Verdict:

PASS.

6.3.1.1.6 FCS_TLSC_EXT.1.3 Guidance

According to the Functional Package, there are no Guidance AA requirements for this SFR.

6.3.1.2 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

6.3.1.2.1 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 section titled **TOE Summary Specifications, FCS_TLS_EXT.2** in the ST to ensure that the description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.

  Upon investigation, the evaluator found that the TSS states that **The TOE supports mutual authentication (MA). It will transmit its client certificate and engage in mutual authentication upon receiving the certificate request message from the server.**

- The evaluator reviewed the TSS section titled **TOE Summary Specifications** in the ST 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.

  Upon investigation, the evaluator found that the TSS states that **Administrators are instructed in the TOE's guidance documentation in order to support MA, administrators must create a client certificate and store the certificate in the appropriate, protected directories. Administrators are also instructed how to configure the invocation of OpenSSL from HTTPS, CLI, and application programs to use the MA arguments in the OpenSSL call. Other than the MA configuration given in the administrator guidance, there is no other steps required to engage in MA**.

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

Verdict:

PASS.

6.3.1.2.2 FCS_TLSC_EXT.2.1 Guidance

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 section titled **TLS Mutual Authentication** in the AGD and found that it includes detailed instructions for configuring the client-side certificates for TLS mutual authentication.

  The AGD states **Archon OS supports optional mutual authentication (MA) communicating with servers. An X.509 device certificate for the TOE must be configured in order to support MA. The TOE will send its client certificate and engage in MA when it sees the certificate request message is sent by the server**.
- The evaluator checked the AGD and ensured that the guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.

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

Verdict:

PASS.

6.3.1.3 **FCS_TLSC_EXT.4 TLS Client Support for Renegotiation**

6.3.1.3.1 FCS_TLSC_EXT.4.1 TSS

According to the Functional Package, there are no TSS AA requirements for this SFR.

6.3.1.3.2 FCS_TLSC_EXT.4.1 Guidance

According to the Functional Package, there are no Guidance AA requirements for this SFR.

6.3.1.4 **FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension**

6.3.1.4.1 FCS_TLSC_EXT.5.1 TSS

Objective:

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

Evaluator Findings:

- The evaluator examined the section titled **TOE Summary Specifications, FCS_TLSC_EXT.5** in the ST to verify that it describes the Supported Groups Extension.

  Upon investigation, the evaluator found that the TSS states that **The TOE presents the supported Elliptic Curves Extension in the Client Hello message with the P-384 curve (secp384r1).**

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

Verdict:

PASS.

6.3.1.4.2 FCS_TLSC_EXT.5.1 Guidance

According to the Functional Package, there are no Guidance AA requirements for this SFR.

7 Evaluation Activities for Security Assurance Requirements

7.1 Class ADV: Development

The information about the OS is contained in the guidance documentation available to the end user as well as the TSS portion of the ST. The OS developer must concur with the description of the product that is contained in the TSS as it relates to the functional requirements. The evaluation activities contained in Section 5.1 Security Functional Requirements should provide the ST authors with sufficient information to determine the appropriate content for the TSS section.

7.1.1 ADV_FSP.1 Basic Functional Specification

The functional specification describes the TSFIs. It is not necessary to have a formal or complete specification of these interfaces. Additionally, because OSes conforming to this PP will necessarily have interfaces to the operational environment that are not directly invokable by OS users, there is little point specifying that such interfaces be described in and of themselves since only indirect testing of such interfaces may be possible. For this PP, the activities for this family should focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation. No additional "functional specification" documentation is necessary to satisfy the evaluation activities specified. The interfaces that need to be evaluated are characterized through the information needed to perform the assurance activities listed, rather than as an independent, abstract list.

Evaluation Activity:

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

Evaluators Findings:

- The provided functional specification documentation was comprehensive and provided sufficient information to support all evaluation activities. As a result, the evaluator was able to effectively carry out all related TSS, AGD, and testing assurance activities.

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

Verdict:

PASS.

7.2 Class AGD: Guidance Documents

The guidance documents will be provided with the ST. Guidance must include a description of how the IT personnel verifies that the operational environment can fulfill its role for the security functionality. The documentation should be in an informal style and readable by the IT personnel. Guidance must be provided for every operational environment that the product supports as claimed in the ST. This guidance includes instructions to successfully install the TSF in that environment; and Instructions to manage the security of the TSF as a product and as a component of the larger operational environment. Guidance pertaining to particular security functionality is also provided; requirements on such guidance are contained in the Evaluation Activities specified with each requirement.

7.2.1 AGD_OPE.1 Operational User Guidance (AGD_OPE.1)

Evaluation Activities:

Some of the contents of the operational guidance are verified by the evaluation activities in Section 5.1 Security Functional Requirements and evaluation of the OS according to the [CEM]. The following additional information is also required.

- If cryptographic functions are provided by the OS, the operational guidance will contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the OS.
- It will provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the OS.
- The documentation must describe the process for verifying updates to the OS by verifying a digital signature – this may be done by the OS or the underlying platform.
- The evaluator will 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 OS (e.g., placement in a specific directory).
    - Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature.
    - The OS will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.

Evaluators Findings:

- The evaluator examined the section titled **Configuring Archon OS into the CC Evaluated Configuration** in the AGD to verify that it contains instructions for configuring the cryptographic engine associated with the evaluated configuration of the OS.

  Upon investigation, the evaluator found that the AGD states that **SCAP support and configuration combined with OSPP support and configuration means that by default, Archon OS v3.0.0.2 is configured with a subset of CC evaluated configuration parameters. Specifically, there are no TLS parameters that need to be configured (with the exception of certificates) and the TOE is automatically configured in FIPS mode.**

- The evaluator examined the section titled **Disclaimers** in the AGD to verify that it provides a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the OS.

  Upon investigation, the evaluator found that the AGD states that **OpenSSL was the only tested cryptographic engine. Other cryptographic engines were not evaluated nor tested, so they should not be used.**

- The evaluator reviewed the AGD's update process and determined: section titled **Update Signature Verification** describes the process of verifying the update's digital signature.
- The evaluator reviewed the AGD and determined that
    - The evaluator reviewed the AGD, section tiled **Secure Acceptance of the TOE**, and verified that the AGD describes the instructions for obtaining the TOE.
    - The evaluator reviewed the AGD, section titled **System Updates** and verified section titled **Installation Prerequisites, Preliminary Setup,** and **Upgrade Process** all describe instructions for initiating the

update process. The evaluator also reviewed section **Update Signature Verification** and determined that the section includes a description of what the administrator should expect if an update's signature is successful or failed.

- The evaluator reviewed the ST and verified if there is any functionality excluded from the scope of the evaluation. Section **Product Functionality not Included in the Scope of the Evaluation** in the ST states that: **SELinux Mandatory Access Control System, OS Virtualization Infrastructure, and Containerization infrastructure** are excluded from the evaluation.

  The evaluator then reviewed the AGD and verified section titled **Product Functionality not Included in the Scope of the Evaluation** identifies the security functionality excluded from the evaluation and that that list matches the list in the ST.

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

Verdict:

PASS.

7.2.2 AGD_PRE.1 Preparative Procedures (AGD_PRE.1)

Evaluation Activity:

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

Evaluator Findings

- The evaluator examined the AGD to verify that it adequately addresses all platforms claimed for the OS in the ST.

  Upon investigation, the evaluator found that the AGD describes all supported hardware platforms in section **Product Overview** and describes the composition of the operational environment in section **Operational Environment.**

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

Verdict:

PASS.

7.3 Class ALC: Life-cycle Support

At the assurance level provided for OSes conformant to this PP, life-cycle support is limited to end-user-visible aspects of the life-cycle, rather than an examination of the OS vendor's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it is a reflection on the information to be made available for evaluation at this assurance level.

7.3.1 ALC_CMC.1 Labeling of the TOE (ALC_CMC.1)

This component is targeted at identifying the OS such that it can be distinguished from other products or versions from the same vendor and can be easily specified when being procured by an end user.

Evaluation Activity:

- The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST.

- Further, the evaluator will check the AGD guidance and OS 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 OS, 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 Security Target to verify that the ST contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST.

  Upon investigation, the evaluator found that the ST provides a product name and version number in section titled **Security Target and TOE Reference.**

- The evaluator examined the AGD to verify that the version number is consistent with that in the ST.

  Upon investigation, the evaluator found that the AGD's Title Page and section titled **Purpose** both identify the TOE version that is consistent with the [ST].

- The evaluator examined the vendor web site to ensure that the information in the ST is sufficient to distinguish the product.

  Upon investigation, the evaluator found that the vendor does not maintain a web site advertising the TOE.

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

Verdict:

PASS.

7.3.2 ALC_CMS.1 TOE CM Coverage (ALC_CMS.1)

Given the scope of the OS and its associated evaluation evidence requirements, this component's evaluation activities are covered by the evaluation activities listed for ALC_CMC.1.

Evaluation Activity:

- The "evaluation evidence required by the SARs" in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the OS is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the evaluation activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component. Life-cycle support is targeted aspects of the developer's life-cycle and instructions to providers of applications for the developer's devices, rather than an in-depth examination of the TSF manufacturer's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it's a reflection on the information to be made available for evaluation.
- The evaluator will ensure that the developer has identified (in guidance documentation for application developers concerning the targeted platform) one or more development environments appropriate for use in developing applications for the developer's platform. For each of these development environments, the developer shall provide information on how to configure the environment to ensure that buffer overflow protection mechanisms in the environment(s) are invoked (e.g., compiler and linker flags). The evaluator will ensure that this documentation also includes an indication of whether such protections are on by default, or have to be specifically enabled.

- The evaluator will 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.

Evaluators findings:

- The "evaluation evidence required by the SARs" is covered by the evaluation activities listed for ALC_CMC.1, above.
- The evaluator examined the platform developer guidance documentation to verify that it identifies one or more development environments appropriate for use in developing applications for the developer's platform. For each of these development environments, the evaluator verified that the developer provides information on how to configure the environment to ensure that buffer overflow protection mechanisms in the environment(s) are invoked (e.g., compiler flags) and whether such protections are on by default.

  Upon investigation, the evaluator found that [ST] section 6 **TOE Summary Specification, FPT_SBOP_EXT.1** states that "The TOE is a stack-based OS and is compiled with the option "stack-protector-strong" to add a stack canary and associated verification code during the entry and exit of function frames to prevent stack-based buffer overflows."
- The evaluator examined the section titled **Introduction** in the AGD to verify that it is associated with the TSF using unique identification.

  Upon investigation, the evaluator found that the guidance documentation states that the TOE is Archon OS version 3.0.0.2, which is consistent with the [ST]. The evaluator verified with the developers that CACI has only one Archon OS product.

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

Verdict:

PASS.

7.3.3 ALC_TSU_EXT.1 Timely Security Updates

This component requires the OS developer, in conjunction with any other necessary parties, to provide information as to how the end-user devices are updated to address security issues in a timely manner. The documentation describes the process of providing updates to the public from the time a security flaw is reported/discovered, to the time an update is released. This description includes the parties involved (e.g., the developer, carriers(s)) and the steps that are performed (e.g., developer testing, carrier testing), including worst case time periods, before an update is made available to the public.

Evaluation Activity:

- The evaluator will verify that the TSS contains a description of the timely security update process used by the developer to create and deploy security updates. The evaluator will verify that this description addresses the entire application. The evaluator will also verify that, in addition to the OS developer's process, any third-party processes are also addressed in the description. The evaluator will also verify that each mechanism for deployment of security updates is described.
- The evaluator will verify that, for each deployment mechanism described for the update process, the TSS lists a time between public disclosure of a vulnerability and public availability of the security update to the OS patching this vulnerability, to include any third-party or carrier delays in deployment. The evaluator will verify that this time is expressed in a number or range of days.

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

Evaluators findings:

- The evaluator examined section titled **TOE Summary Specification, ALC_TSU_EXT.1** of the Security Target and found that upon investigation, the evaluator found that the TSS states **Archon OS vulnerabilities may be identified via internal testing, monitoring of CVE reports for Archon OS and third-party components, notification of vulnerabilities from third-party suppliers, or from customer reports**. Additionally, the TSS states **Customers are notified when releases are available, and provided with a URL for download.**
- The evaluator examined section titled **TOE Summary Specification, ALC_TSU_EXT.1** of the Security Target and found that upon investigation, the evaluator found that the TSS states **CACI provides a security update release for Archon OS at least once every 3 months. Resolution of vulnerabilities is expected within 180 days of public disclosure. For significant vulnerabilities, additional releases may be generated for quicker resolution.**
- The evaluator examined the section titled **TOE Summary Specification, ALC_TSU_EXT.1** in the Security Target to verify that the TSS includes the publicly available mechanisms for reporting security issues related to the TOE, including a method for protecting the report.

  Upon investigation, the evaluator found that the TSS states **Customers may report security issues related to Archon OS via the secure support portal at https://attilasec.zendesk.com/hc/en-us/requests/new [attilasec.zendesk.com].** Additionally, the TSS states, **For vulnerabilities involving CACI-developed components, the CACI engineering team creates a Github ticket for each vulnerability to track the analysis and resolution of the issue. Issues are prioritized and worked to resolution, then incorporated into a product release.**

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

Verdict:

PASS.

7.4 Class ATE: Tests

7.4.1 ATE_IND.1 Independent Testing – Conformance

Testing is performed to confirm the functionality described in the TSS as well as the administrative (including configuration and operational) documentation provided. The focus of the testing is to confirm that the requirements specified in Section 5.1 Security Functional Requirements being met, although some additional testing is specified for SARs in Section 5.2 Security Assurance Requirements. The evaluation activities identify the additional testing activities associated with these components. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/OS combinations that are claiming conformance to this PP. Given the scope of the OS and its associated evaluation evidence requirements, this component's evaluation activities are covered by the evaluation activities listed for ALC_CMC.1.

Evaluation Activity:

- The evaluator will prepare a test plan and report documenting the testing aspects of the system, including any application crashes during testing. The evaluator will 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 affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary. The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the OS and its platform.
- This also includes the configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by this PP and used by the cryptographic protocols being evaluated (IPsec, TLS). 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 will 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:

- The evaluator constructed and executed a test plan, which was submitted as part of this evaluation as a test report document. During testing, no application crashes were reported. The evaluator ensured that all testing actions outlined in the Common Evaluation Methodology (CEM) and the body of the Protection Profile evaluation activities were represented within the test plan and suitably replicated within this public-facing document.
- The evaluator’s test plan provided full coverage of the applicable testing requirements. The test plan has a section devoted to the platforms under test and any equivalency arguments to justify platforms which were not explicitly tested. The test plan further provides information on the testing environment including setup necessary beyond the AGD documentation.
- The AGD instructs that no specific configuration of the cryptographic engine in use is required and therefore, the test plan did not require configuration information. The test plan includes high-level test objectives and test procedures to be followed to achieve those objections. Additionally, the test plan includes expected test results.

  The test plan includes high level objectives, which are replicated within this public facing document, and the test steps needed to achieve that objective.
- The test plan includes the expected results and includes test evidence that provides the actual results.

7.5 Class AVA: Vulnerability Assessment

For the first generation of this protection profile, the evaluation lab is expected to survey open sources to discover what vulnerabilities have been discovered in these types of products. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. Until penetration tools are created and uniformly distributed to the evaluation labs, the evaluator will not be expected to test for these vulnerabilities in the OS. The labs will be expected to comment on the likelihood of these vulnerabilities given the documentation provided by the vendor. This information will be used in the development of penetration testing tools and for the development of future protection profiles.

7.5.1 AVA_VAN.1 Vulnerability Survey (AVA_VAN.1)

For the first generation of this protection profile, the evaluation lab is expected to survey open sources to discover what vulnerabilities have been discovered in these types of products. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. Until penetration tools are created and uniformly distributed to the evaluation labs, the evaluator will not be expected to test for these vulnerabilities in the OS. The labs will be expected to comment on the likelihood of these vulnerabilities given the documentation provided by the vendor. This information will be used in the development of penetration testing tools and for the development of future protection profiles.

Evaluation Activity:

- The evaluator will 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.

Evaluator Findings:

- The evaluator examined sources of publicly available information to identify potential vulnerabilities in the TOE. The sources examined are as follows:
    - **https://nvd.nist.gov/view/vuln.search**
    - **http://cve.mitre.org/cve**
    - **https://www.cvedetails.com/vulnerability-search.php**
    - **https://www.kb.cert.org/vuls/search/**
    - **www.exploitsearch.net**
    - **www.securiteam.com**
    - **http://nessus.org/plugins/index.php?view=search**
    - **http://www.zerodayinitiative.com/advisories**
    - **https://www.exploit-db.com**
    - **https://www.rapid7.com/db/vulnerabilities**

    The evaluator examined public domain vulnerability searches by performing a keyword search. The terms used for this search were based on the vendor's name, product name, and key platform features leveraged by the product. As a result, the evaluator performed a search using the following keywords:

    - CACI
    - archon-os
    - archon
    - Red Hat Enterprise Linux 8.10
    - aide-0.16-14.el8_5.1s

- TLSV1.2
    - audit-libs-3.1.2-1.el8
    - chrony-4.5-1.el8
    - cryptsetup-libs-2.3.7-7.el8
    - curl-7.61.1-34.el8
    - dnf-4.7.0-20.el8
    - fapolicyd-1.3.2-1.el8
    - firewalld-0.9.11-4.el8
    - gpgme-1.13.1-12.el8
    - grub2-common-2.02-156.el8
    - gnutls-3.6.16-8.el8_9.3
    - gzip-1.9-13.el8_5
    - iptables-1.8.5-11.el8_9
    - kernel-4.18.0-533.el8_10
    - libcap-2.48-6.el8_9.
    - libcap-ng-0.7.11-1.el8
    - libpcap-1.9.1-5.el8
    - openldap-2.4.46-18.el8
    - openssh-8.0p1-24.el8.
    - openssl-1.1.1k-12.el8_9.
    - ostree-libs-2022.2-8.el8.
    - pam-1.3.1-33.el8.x86_64
    - polkit-0.115-15.el8_10.2.
    - rpm-4.14.3-31.el8.x86_64
    - rsyslog-8.2102.0-15.el8.
    - sudo-1.9.5p2-1.el8_9.
    - tar-1.30-9.el8
    - xz-5.2.4-4.el8_6
    - zlib-1.2.11-25.el8

The vulnerability search was performed on 06/17/2024.

- The evaluation lab examined each result provided by the NVD and Red Hat Security Advisory websites to determine if the current TOE version or components within the environment were vulnerable. Based on the analysis, any identified vulnerabilities were patched in the TOE version or prior versions.

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

Verdict:

PASS.

8 NIAP Policy 5

To demonstrate that all cryptographic requirements are satisfied, the Assurance Activity Report must clearly indicate all SFRs for which a CAVP certificate is claimed and include, at a minimum, the cryptographic operation, the NIST standard, the SFR supported, the CAVP algorithm list name (e.g. AES, KAS, CVL, etc.) and the CAVP Certificate number.

This section provides a table that lists all SFRs for which a CAVP certificate is claimed, the CAVP algorithm list name and the CAVP Certificate number.

Table 8: SFR to CAVP Mappings

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
FCS_CKM.1	RSA schemes using cryptographic key sizes of 3072-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3,	OpenSSL version 1.1.1k	RSA KeyGen (FIPS186-4)	A5342
	ECC schemes using "NIST curves" P-384 and [no other curves] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4,	OpenSSL version 1.1.1k	ECDSA KeyGen (FIPS186-4) ECDSA KeyVer (FIPS186-4)	A5342
	FFC Schemes using [safe primes that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes"]	OpenSSL version 1.1.1k	No CAVP certificate. Also, the evaluator confirmed that there is no assurance activity to be performed since "FIPS PUB 186-4" is not selected	
FCS_CKM.2	Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography",	OpenSSL version 1.1.1k	KAS-ECC-SSC SP800-56AR3	A5342
	Finite field-based key establishment schemes that meets NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	OpenSSL version 1.1.1k	KAS-FFC-SSC SP800-56AR3	CCTL has performed all assurance/ evaluation activities and documented in the ETR and AAR accordingly.
FCS_COP.1/ENCRYPT	AES-CBC (as defined in NIST SP 800-38A) and cryptographic key sizes [256-bit]	OpenSSL version 1.1.1k	AES-CBC	A5342

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
	AES-GCM (as defined in NIST SP 800-38D)] and cryptographic key sizes 256-bit and [no other bit size]	OpenSSL version 1.1.1k	AES-GCM	A5342
FCS_COP.1/SIGN	RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4,	OpenSSL version 1.1.1k	RSA SigGen (FIPS186-4)	A5342
			RSA SigVer (FIPS186-4)	A5342
	ECDSA schemes using "NIST curves" P-384 and [no other curves] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5	OpenSSL version 1.1.1k	ECDSA SigGen (FIPS186-4)	A5342
			ECDSA SigVer (FIPS186-4)	A5342
FCS_COP.1/HASH	Cryptographic algorithm [SHA-256,SHA-384,SHA-512] and message digest sizes [• 256 bits, • 384 bits, • 512 bits] that meet the following: [FIPS Pub 180-4].	OpenSSL version 1.1.1k	SHA2-256 SHA2-384 SHA2-512	A5342
		Linux Kernel Crypto API version 4.18.0	SHA2-512	A5343
FCS_COP.1/KEYHMAC	Cryptographic algorithm [SHA-256, SHA-384, SHA-512] with key sizes [*256 bits, 384 bits, 512 bits*] and message digest sizes [• 256 bits, • 384 bits, • 512 bits] that meet the following: [FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard].	OpenSSL version 1.1.1k	HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512	A5342
		Linux Kernel Crypto API version 4.18.0	HMAC-SHA2-512	A5343

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
FCS_RBG_EXT.1/OSSL	Random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [• CTR_DRBG (AES)].	OpenSSL version 1.1.1k	Counter DRBG(AES-256)	A5342
FCS_RBG_EXT.1/KERN	Random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [• HMAC_DRBG (any)].	Linux Kernel Crypto API version 4.18.0	HMAC DRBG(SHA2-512)	A5343

9 Detailed Test Cases (Test Activities)

9.1 FAU

9.1.1 FAU_GEN.1 Test#1 (TD0693)

Item	Data
Test Assurance Activity	The evaluator will test the OS's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the ST. This should include all instance types of an event specified. When verifying the test results, the evaluator will ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record provide the required information.
Test Steps	1. Audit record for: Start-up and shutdown of the audit functions. • Startup • Shutdown 2. Audit record for: Authentication events (Success/Failure) • Success • Failure 3. Audit record for: Use of privileged/special rights events (Successful and unsuccessful security, audit and configuration changes) • Security Changes ○ Success ○ Failure • Audit Changes ○ Success ○ Failure • Configuration Changes ○ Success ○ Failure 4. Audit record for: Privilege or role escalation events (Success/Failure) • Success • Failure 5. Audit record for: File and object events (Successful and unsuccessful attempts to create, access, delete, modify, modify permissions) • Create ○ Success ○ Failure • Access ○ Success ○ Failure • Delete ○ Success ○ Failure • Modification ○ Success ○ Failure • Permission Change ○ Success

	○ Failure 6. Audit record for: User and Group management events (Successful and unsuccessful add, delete, modify, disable, enable, and credential change) *Note: Enable/Disable functionality is not available for Groups.* • Add User ○ Success ○ Failure • Delete User ○ Success ○ Failure • Modify User ○ Success ○ Failure • Disable User ○ Success ○ Failure • Enable User ○ Success ○ Failure • User Credential Change ○ Success ○ Failure • Add Group ○ Success ○ Failure • Delete Group ○ Success ○ Failure • Modify Group ○ Success ○ Failure • Group Credential Change ○ Success ○ Failure 7. Audit record for: Audit and log data access events (Success/Failure) • Success • Failure 8. Audit record for: Attempted application invocation with arguments (Success/Failure e.g. due to software restriction policy) • Success • Failure 9. Audit record for: System reboot, restart, and shutdown events (Success/Failure) • System reboot ○ Success ○ Failure • System shutdown ○ Success ○ Failure

	10. Audit record for: Kernel module loading and unloading events (Success/Failure) • Module Loading ○ Success ○ Failure • Module Unloading ○ Success ○ Failure 11. Audit record for: Administrator or root-level access events (Success/Failure) • Success • Failure
Expected Test Results	All audit records for events are generated, and match the format specified in the AGD.
Pass/Fail with Explanation	Pass. The TOE generates the appropriate audit logs for each audit event listed in the ST. This satisfies the testing requirement.

9.2 FCS

9.2.1 FCS_CKM.1 Test#1 (TD0712)

Item	Data
Test Assurance Activity	The following content should be included if: • RSA schemes is selected from FCS_CKM.1.1 **Key Generation for FIPS PUB 186-4 RSA Schemes** The evaluator will verify the implementation of RSA Key Generation by the OS 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 will 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 will have the TSF generate 10 keys pairs for each supported key length nlen and verify:

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

Key Generation for Elliptic Curve Cryptography (ECC)
FIPS 186-4 ECC Key Generation Test

For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator will 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 will submit the generated key pairs to the public key verification (PKV) function of a known good implementation.

FIPS 186-4 Public Key Verification (PKV) Test

For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator will 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 will obtain in response a set of 10 PASS/FAIL values.

Key Generation for Finite-Field Cryptography (FFC)
The evaluator will 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

	o Primes q and field prime p shall both be probable primes and two ways to generate the cryptographic group generator g: • Cryptographic Group Generator: o Generator g constructed through a verifiable process o Generator g constructed through an unverifiable process The Key generation specifies 2 ways to generate the private key x: • Private Key: o len(q) bit output of RBG where 1 ≤ x ≤ q-1 o len(q) + 64 bit output of RBG, followed by a mod q-1 operation where 1 ≤ x ≤ q-1 The security strength of the RBG must be at least that of the security offered by the FFC parameter set. To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator must seed the TSF parameter generation routine with sufficient data to deterministically generate the parameter set. For each key length supported, the evaluator will have the TSF generate 25 parameter sets and key pairs. The evaluator will verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. Verification must also confirm: • g != 0,1 • q divides p-1 • g^q mod p = 1 • g^x mod p = y for each FFC parameter set and key pair.
Test Steps	**Key Generation for FIPS PUB 186-4 RSA Schemes** The evaluator examined the ST and found that in Section 6.2 that the TOE was awarded the CAVP certificate #A5342 for FIPS186-4 RSA key generation using the key size of 3072 and 4096 bit. 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 6.2 that the TOE was awarded the CAVP certificate #A5342 for FIPS186-4 ECDSA key generation and verification using P-384 curve. This certificate provides assurance that the TSF performs these functions as required. **Key Generation for Finite-Field Cryptography (FFC)** For FFC Schemes using "safe-prime" there is no CAVP certificate. Additionally, the evaluator confirmed that there is no assurance activity to be performed since "FIPS PUB 186-4" is not selected in the SFR:FCS_CKM.1.
Pass/Fail with Explanation	Pass. The evaluator confirmed the following: For Key Generation for FIPS PUB 186-4 RSA Schemes testing is satisfied by CAVP certificate #A5342.

	For Key Generation and verification for Elliptic Curve Cryptography (ECC) testing is satisfied by CAVP certificate #A5342. For FFC Schemes using "safe-prime" there is no CAVP certificate. Additionally, the evaluator confirmed that there is no assurance activity to be performed since "FIPS PUB 186-4" is not selected in the SFR:FCS_CKM.1.1.

9.2.2 FCS_CKM.2

Item	Data
Test Assurance Activity	**Key Establishment Schemes** The evaluator will verify the implementation of the key establishment schemes supported by the OS using the applicable tests below. *The following content should be included if:* - Elliptic curve-based, Finite field-based is selected from FCS_CKM.2.1 **SP800-56A Key Establishment Schemes** The evaluator will verify the OS'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 the OS has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the discrete logarithm cryptography (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 will 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 MAC data and the calculation of MAC tag. **Function Test** The Function test verifies the ability of the OS to implement the key agreement schemes correctly. To conduct this test the evaluator will generate or obtain test vectors from a known good implementation of the OS's supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of 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 will obtain the DKM, the corresponding OS's public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and OS id fields.

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

The evaluator will 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 OS shall perform the above for each implemented approved MAC algorithm.

Validity Test

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

The evaluator will inject an error in some of the test vectors to test that the OS recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MAC'd, or the generated MAC tag. If the OS 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 OS's static private key to assure the OS 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 OS shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator will compare the OS's results with the results using a known good implementation verifying that the OS detects these errors.

The following content should be included if:

- *RSA-based is selected from FCS_CKM.2.1*

RSAES-PKCS1-v1_5 Key Establishment Schemes

	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_ITC_EXT.1 that uses RSAES-PKCS1-v1_5. *The following content should be included if:* • *Finite field-based is selected from FCS_CKM.2.1* **FFC Schemes using "safe-prime" groups (identified in Appendix D of SP 800-56A Revision 3)** 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_ITC_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 #A5342 for Elliptic curve-based key establishment schemes that meets NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography. This certificate provides assurance that the TSF performs these functions as required. **RSAES-PKCS1-v1_5 Key Establishment Schemes** The TOE does not use or claim RSAES-PKCS1-v1_5 key establishment schemes. **FFC Schemes using "safe-prime" groups (identified in Appendix D of SP 800-56A Revision 3)** In test case FCS_TLSC_EXT.2.1 Test #2, the valuator verified the correctness of the TSF's implementation of "safe-prime" groups by using a known good implementation for each protocol selected in FTP_ITC_EXT.1, which are the update server and user-initiated TLS application that uses "safe-prime" groups. On both cases, the evaluator determined that the DHE key seize was 3072bits (384 bytes) used by the TOE and the server and that the server used the NIST prime P-3072 Group in the server key exchange TLS message.
Pass/Fail with Explanation	**SP800-56A Key Establishment Schemes** For Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3 Pass: CAVP Cert #A5342. shows the TOE correctly implements ECDHE key agreement using P-384. **RSAES-PKCS1-v1_5 Key Establishment Schemes** N/A, because the TOE does not use or claim RSAES-PKCS1-v1_5 key establishment schemes. For **FFC Schemes using "safe-prime" groups (identified in Appendix D of SP 800-56A Revision 3)**

	Pass: the correctness of the TSF's implementation of "safe-prime" groups by using a known good implementation for each protocol selected in FTP_ITC_EXT.1 that uses "safe-prime" groups.

9.2.3 FCS_CKM_EXT.4 Test#1

Item	Data
Test Assurance Activity	Test 1: Applied to each key held as in volatile memory and subject to destruction by overwrite by the TOE (whether or not the value is subsequently encrypted for storage in volatile or non-volatile memory). In the case where the only selection made for the destruction method key was removal of power, then this test is unnecessary. The evaluator will: 1. Record the value of the key in the TOE subject to clearing. 2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1. 3. Cause the TOE to clear the key. 4. Cause the TOE to stop the execution but not exit. 5. Cause the TOE to dump the entire memory of the TOE into a binary file. 6. Search the content of the binary file created in Step #5 for instances of the known key value from Step #1 Steps 1-6 ensure that the complete key does not exist anywhere in volatile memory. If a copy is found, then the test fails.
Test Steps	1. Start debugging the OpenSSL file on the TOE. 2. Set Breakpoints and initiate a connection to TLS server: 3. Reach Breakpoint 1 and print keys. 4. Print next keys and see the keys stored on relevent memory location. 5. Continue to next breakpoint. 6. Print key values and show them saved in relevant memory address. 7. Continue to last breakpoint. 8. Reach the end of breakpoint. 9. Verify that the keys were zeroized by comparing memory addresses and quit GDB. 10. Move gcore-dump files to test VM and convert to '.hex' files. 11. Search for the following key values in all .hex files and verify that the keys are not found in zeroized file.
Expected Test Results	The TOE should properly destroy keys.
Pass/Fail with Explanation	Pass. The TOE properly destroys keys. This satisfies the testing requirement.

9.2.4 FCS_CKM_EXT.4 Test#2

Item	Data
Test Assurance Activity	Test 2: Applied to each key help in non-volatile memory and subject to destruction by the TOE. The evaluator will use special tools (as needed), provided by the TOE developer if necessary, to ensure the tests function as intended.

	1. Identify the purpose of the key and what access should fail when it is deleted. (e.g. the data encryption key being deleted would cause data decryption to fail.) 2. Cause the TOE to clear the key. 3. Have the TOE attempt the functionality that the cleared key would be necessary for. The test succeeds if step 3 fails.
Test Steps	1. The evaluators started a TLS session on a server. 2. The evaluator attempted a successful TLS connection from the TOE using Root Certificate. 3. Verify the connection was successful and the certificate was used. 4. Delete the Root certificate from the directory. 5. Verify the connection fails when TOE tries to connect to the TLS server.
Expected Test Results	The TOE should not be allowed to perform a function that relies on keys that have been removed.
Pass/Fail with Explanation	Pass. The TOE behaves as expected when the cryptographic keys are removed from the non-volatile memory. This satisfies the test requirements

9.2.5 FCS_CKM_EXT.4 Test#3

Item	Data
Test Assurance Activity	**Tests 3 and 4** do not apply for the selection **instructing the underlying platform to destroy the representation of the key**, as the TOE has no visibility into the inner workings and completely relies on the underlying platform. **Test 3:** The following tests are used to determine the TOE is able to request the platform to overwrite the key with a TOE supplied pattern. Applied to each key held in non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator will use a tool that provides a logical view of the media (e.g., MBR file system): 1. Record the value of the key in the TOE subject to clearing. 2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1. 3. Cause the TOE to clear the key. 4. Search the logical view that the key was stored in for instances of the known key value from Step #1. If a copy is found, then the test fails.
Test Steps	1. Show Root_CA.crt present on TOE. 2. Initiate a TLSC session using the same certificate chain. 3. Shred the Root_CA.crt certificate. 4. Show the value of certificate after shredding. 5. Search for key value in ICA1.pem and verify it is not found. 6. Try a TLSC connection and verify it fails.
Expected Test Results	The TOE should be able to request the platform to overwrite the key.
Pass/Fail with Explanation	Pass. TOE was able to request the platform to overwrite the key. This satisfies the testing requirements.

9.2.6 FCS_CKM_EXT.4 Test#4

Item	Data
Test Assurance Activity	**Tests 3 and 4** do not apply for the selection **instructing the underlying platform to destroy the representation of the key**, as the TOE has no visibility into the inner workings and completely relies on the underlying platform. **Test 4:** Applied to each key held as non-volatile memory and subject to destruction by overwrite by the TOE. The evaluator will use a tool that provides a logical view of the media: 1. Record the logical storage location of the key in the TOE subject to clearing. 2. Cause the TOE to perform a normal cryptographic processing with the key from Step #1. 3. Cause the TOE to clear the key. 4. Read the logical storage location in Step #1 of non-volatile memory to ensure the appropriate pattern is utilized. The test succeeds if correct pattern is used to overwrite the key in the memory location. If the pattern is not found the test fails.
Pass/Fail with Explanation	Pass. The testing requirements are satisfied by FCS_CKM_EXT.4Test#3.

9.2.7 FCS_COP.1/ENCRYPT (TD0712)

Item	Data
Test Assurance Activity	The following content should be included if: • AES-XTS is selected from FCS_COP.1.1/ENCRYPT **XTS-AES Test** The evaluator will test the encrypt functionality of XTS-AES for each combination of the following input parameter lengths: • 512 bit (for AES-256) key • Three data unit (i.e., plaintext) lengths. One of the data unit lengths will be a nonzero integer multiple of 256 bits, if supported. One of the data unit lengths will be an integer multiple of 256 bits, if supported. The third data unit length will be either the longest supported data unit length or 216 bits, whichever is smaller. using a set of 100 (key, plaintext and 256-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 will 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 XTSAES decrypt. The following content should be included if: • AES-CBC is selected from FCS_COP.1.1/ENCRYPT

AES-CBC Known Answer Tests

There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values will be 256-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 will compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

- Test 5: To test the encrypt functionality of AES-CBC, the evaluator will supply a set of 5 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. The plaintext values will encrypted with a 256-bit all-zeros key. To test the decrypt functionality of AES-CBC, the evaluator will perform the same test as for encrypt, using 5 ciphertext values as input and AES-CBC decryption.
- Test 6: To test the encrypt functionality of AES-CBC, the evaluator will supply a set of five 256-keys 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. To test the decrypt functionality of AES-CBC, the evaluator will perform the same test as for encrypt, using an all-zero ciphertext value as input and AES-CBC decryption.
- Test 7: To test the encrypt functionality of AES-CBC, the evaluator will supply the a 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. Key i will 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 will supply the set 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 set of key/ciphertext pairs will have 256 256-bit key/ciphertext pairs. Key i in each set will 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 will be the value that results in an all-zeros plaintext when decrypted with its corresponding key.
- Test 8: To test the encrypt functionality of AES-CBC, the evaluator will supply the set of 256 plaintext values described below and obtain the ciphertext values that result from AES-CBC encryption of the given plaintext using a 256-bit key value of all zeros with an IV of all zeros. Plaintext value i in each set will have the leftmost i bits be ones and the rightmost 256-i bits be zeros, for i in [1,256].

To test the decrypt functionality of AES-CBC, the evaluator will 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 will test the encrypt functionality by encrypting an i-block message where 1 < i ≤ 10. The evaluator will 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 will be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation. The evaluator will also test the decrypt functionality for each mode by decrypting an i-block message where 1 < i ≤10. The evaluator will 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 will 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 will test the encrypt functionality using a set of 100 plaintext, IV, and key 3- tuples. The keys, plaintext, and IV values are each 256-bits. For each 3-tuple, 1000 iterations will be run as follows:

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

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

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

The following content should be included if:

- AES-CTR is selected from FCS_COP.1.1/ENCRYPT

AES-CTR Test

Known Answer Tests (KATs)

There are four Known Answer Tests (KATs) described below. For all KATs, the plaintext, initialization vector (IV), and ciphertext values shall be 256-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 will compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

- Test 9: To test the encrypt functionality, the evaluator will supply 5 plaintext values and obtain the ciphertext value that results from encryption of the given plaintext using a 256-bit key value of all zeros and an IV of all zeros. To test the decrypt functionality, the evaluator will perform the same test as for encrypt, using the 5 ciphertext values as input.

- Test 10: To test the encrypt functionality, the evaluator will supply 5 256-bit 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. To test the decrypt functionality, the evaluator will perform the same test as for encrypt, using an all zero ciphertext value as input.
- Test 11: To test the encrypt functionality, the evaluator will supply a set 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 and an IV of all zeros. The set of keys shall have shall have 256 256-bit keys. Keyi shall have the leftmost i bits be ones and the rightmost 256-i bits be zeros, for i in [1, N]. To test the decrypt functionality, the evaluator will supply the set 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 set of key/ciphertext pairs shall have 256 256-bit pairs. Keyi shall have the leftmost i bits be ones and the rightmost 256-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.
- Test 12: To test the encrypt functionality, the evaluator will supply the set of 256 plaintext values described below and obtain the two ciphertext values that result from encryption of the given plaintext 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 256-i bits be zeros, for i in [1, 256]. To test the decrypt functionality, the evaluator will perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input.

Multi-Block Message Test

The evaluator will test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10. For each i the evaluator will 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 will 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 will 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.

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 will test the encrypt functionality using 100 plaintext/key pairs. Each key shall be 256-bit. The plaintext values shall be 256-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.

The following content should be included if:

- AES Key Wrap (KW) (as defined in NIST SP 800-38F), AES Key Wrap with Padding (KWP) (as defined in NIST SP 800-38F) is selected from FCS_COP.1.1/ENCRYPT

AES Key Wrap (AES-KW) and Key Wrap with Padding (AES-KWP) Test

The evaluator will test the authenticated encryption functionality of AES-KW for EACH combination of the following input parameter lengths:

- 256 bit key encryption keys (KEKs)
- Three plaintext lengths. One of the plaintext lengths will be two semi-blocks (256 bits). One of the plaintext lengths will be three semi-blocks (192 bits). The third data unit length will be the longest supported plaintext length less than or equal to 64 semi-blocks (4096 bits).

using a set of 100 key and plaintext pairs and obtain the ciphertext that results from AES-KW authenticated encryption. To determine correctness, the evaluator will use the AES-KW authenticated-encryption function of a known good implementation.

The evaluator will test the authenticated-decryption functionality of AES-KW using the same test as for authenticated-encryption, replacing plaintext values with ciphertext values and AES-KW authenticated-encryption with AES-KW authenticated-decryption.

The following content should be included if:

- AES Key Wrap with Padding (KWP) (as defined in NIST SP 800-38F) is selected from FCS_COP.1.1/ENCRYPT

The evaluator will test the authenticated-encryption functionality of AES-KWP using the same test as for AES-KW authenticated-encryption with the following change in the three plaintext lengths:

- One plaintext length will be one octet. One plaintext length will be 20 octets (160 bits).
- One plaintext length will be the longest supported plaintext length less than or equal to 512 octets (4096 bits).

The evaluator will test the authenticated-decryption functionality of AES-KWP using the same test as for AES-KWP authenticated-encryption, replacing plaintext values with ciphertext values and AES-KWP authenticated-encryption with AES-KWP authenticated-decryption.

The following content should be included if:

- AES-CCM or AES-CCMP-256 is selected from FCS_COP.1.1/ENCRYPT

AES-CCM Tests

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

- 128 bit (if selected) and 256 bit keys
- Two payload lengths. One payload length will be the shortest supported payload length, greater than or equal to zero bytes. The other payload length will be the longest supported payload length, less than or equal to 32 bytes (256 bits).
- Two or three associated data lengths. One associated data length will be 0, if supported. One associated data length will be the shortest supported payload length, greater than or equal to zero bytes. One associated data length will be the longest supported payload length, less than or equal to 32 bytes (256 bits). If the implementation supports an associated data length of 2 16 bytes, an associated data length of 216 bytes will be tested.
- Nonce lengths. The evaluator will test all nonce lengths between 7 and 13 bytes, inclusive, that are supported by the OS.
- Tag lengths. The evaluator will test all of the following tag length values that are supported by the OS: 4, 6, 8, 10, 12, 14 and 16 bytes.

To test the generation-encryption functionality of AES-CCM, the evaluator will perform the following four tests:

- Test 13: For EACH supported key and associated data length and ANY supported payload, nonce and tag length, the evaluator will supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.
- Test 14: For EACH supported key and payload length and ANY supported associated data, nonce and tag length, the evaluator will supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.
- Test 15: For EACH supported key and nonce length and ANY supported associated data, payload and tag length, the evaluator will supply one key value and 10 associated data,

payload and nonce value 3-tuples and obtain the resulting ciphertext.

- Test 16: For EACH supported key and tag length and ANY supported associated data, payload and nonce length, the evaluator will supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

To determine correctness in each of the above tests, the evaluator will compare the ciphertext with the result of generation-encryption of the same inputs with a known good implementation.

To test the decryption-verification functionality of AES-CCM, for EACH combination of supported associated data length, payload length, nonce length and tag length, the evaluator will supply a key value and 15 nonce, associated data and ciphertext 3-tuples and obtain either a FAIL result or a PASS result with the decrypted payload. The evaluator will supply 10 tuples that should FAIL and 5 that should PASS per set of 15.

Additionally, the evaluator will use tests from the IEEE 802.11-02/362r6 document "Proposed Test vectors for IEEE 802.11 TGi", dated September 10, 2002, Section 2.1 AESCCMP Encapsulation Example and Section 2.2 Additional AES CCMP Test Vectors to further verify the IEEE 802.11-2007 implementation of AES-CCMP.

The following content should be included if:

- AES-GCMP-256 is selected from FCS_COP.1.1/ENCRYPT

AES-GCMP Test

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

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

The evaluator will 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 will 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 will 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 will 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 will compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

AES-GCMP Monte Carlo Tests

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

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

The evaluator will 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 will 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 will 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 will 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 will compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

The following content should be included if:

AES-CCM or AES-CCMP-256 is selected from FCS_COP.1.1/ENCRYPT

AES-CCM Tests

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

- 128 bit (if selected) and 256 bit keys
- Two payload lengths. One payload length will be the shortest supported payload length, greater than or equal to zero bytes. The other payload length will be the longest supported payload length, less than or equal to 32 bytes (256 bits).
- Two or three associated data lengths. One associated data length will be 0, if supported. One associated data length will be the shortest supported payload length, greater than or equal to zero bytes. One associated data length will be the longest supported payload length, less than or equal to 32 bytes (256 bits). If the implementation supports an associated data length of 2 16 bytes, an associated data length of 216 bytes will be tested.
- Nonce lengths. The evaluator will test all nonce lengths between 7 and 13 bytes, inclusive, that are supported by the OS.
- Tag lengths. The evaluator will test all of the following tag length values that are supported by the OS: 4, 6, 8, 10, 12, 14 and 16 bytes.

To test the generation-encryption functionality of AES-CCM, the evaluator will perform the following four tests:

· Test 13: For EACH supported key and associated data length and ANY supported payload, nonce and tag length, the evaluator will supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

· Test 14: For EACH supported key and payload length and ANY supported associated data, nonce and tag length, the evaluator will supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

· Test 15: For EACH supported key and nonce length and ANY supported associated data, payload and tag length, the evaluator will supply one key value and 10 associated data, payload and nonce value 3-tuples and obtain the resulting ciphertext.

· Test 16: For EACH supported key and tag length and ANY supported associated data, payload and nonce length, the evaluator will supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

To determine correctness in each of the above tests, the evaluator will compare the ciphertext with the result of generation-encryption of the same inputs with a known good implementation.

To test the decryption-verification functionality of AES-CCM, for EACH combination of supported associated data length, payload length, nonce

	length and tag length, the evaluator will supply a key value and 15 nonce, associated data and ciphertext 3-tuples and obtain either a FAIL result or a PASS result with the decrypted payload. The evaluator will supply 10 tuples that should FAIL and 5 that should PASS per set of 15. Additionally, the evaluator will use tests from the IEEE 802.11-02/362r6 document "Proposed Test vectors for IEEE 802.11 TGi", dated September 10, 2002, Section 2.1 AESCCMP Encapsulation Example and Section 2.2 Additional AES CCMP Test Vectors to further verify the IEEE 802.11-2007 implementation of AES-CCMP.
Test Steps	**XTS-AES Test** This is not applicable as the TOE does not claim or use AES in XTS mode. **AES-CBC Known Answer Tests** The evaluator examined the ST and found that in Section 6.2 that the TOE was awarded the CAVP certificate #A5342 for AES-CBC (NIST SP800-38A) using key size 256 for encryption and decryption. This certificate provides assurance that the TSF performs these functions as required. **AES-CBC Multi-Block Message Test** The evaluator examined the ST and found that in Section 6.2 that the TOE was awarded the CAVP certificate #A5342 for AES-CBC (NIST SP800-38A) using key size 256 for encryption and decryption. This certificate provides assurance that the TSF performs these functions as required. **AES-CBC Monte Carlo Tests** The evaluator examined the ST and found that in Section 6.2 that the TOE was awarded the CAVP certificate #A5342 for AES-CBC (NIST SP800-38A) using key size 256 for encryption and decryption. This certificate provides assurance that the TSF performs these functions as required. **AES-CTR Test Known Answer Tests (KATs)** This is not applicable as the TOE does not claim or use AES in CTR mode. **AES-CTR Multi-Block Message Test** This is not applicable as the TOE does not claim or use AES in CTR mode. **AES-CTR Monte-Carlo Test** This is not applicable as the TOE does not claim or use AES in CTR mode. **AES Key Wrap (AES-KW) and Key Wrap with Padding (AES-KWP) Test** This is not applicable as the TOE does not claim or use AES in KW/KWP mode. **AES-CCM Tests** This is not applicable as the TOE does not claim or use AES in CCM mode. **AES-GCMP Test** This is not applicable as the TOE does not claim or use AES-GCMP mode. **AES-GCMP Monte Carlo Tests** This is not applicable as the TOE does not claim or use AES-GCMP Monte Carlo Tests. **AES-CCM Tests** This is not applicable as the TOE does not claim or use AES-CCM mode. **AES-GCM Tests** The OSPPv4.3 does not include tests for AES-GCM; however, CAVP certificate #5342 verifies that the TOE correctly implements AES-GCM.

Pass/Fail with Explanation	**XTS-AES Test** N/A. This is not applicable as the TOE does not claim or use AES in XTS mode. **AES-CBC Known Answer Tests** Pass. The evaluator examined the ST and found that in Section 6.2 that the TOE was awarded the CAVP certificate #A5342 for AES-CBC (NIST SP800-38A) using key size 256 for encryption and decryption. This certificate provides assurance that the TSF performs these functions as required. **AES-CBC Multi-Block Message Test** Pass. The evaluator examined the ST and found that in Section 6.2 that the TOE was awarded the CAVP certificate #A5342 for AES-CBC (NIST SP800-38A) using key size 256 for encryption and decryption. This certificate provides assurance that the TSF performs these functions as required. **AES-CBC Monte Carlo Tests** Pass. The evaluator examined the ST and found that in Section 6.2 that the TOE was awarded the CAVP certificate #A5342 for AES-CBC (NIST SP800-38A) using key size 256 for encryption and decryption. This certificate provides assurance that the TSF performs these functions as required. **AES-CTR Test Known Answer Tests (KATs)** N/A. This is not applicable as the TOE does not claim or use AES in CTR mode. **AES-CTR Multi-Block Message Test** N/A. This is not applicable as the TOE does not claim or use AES in CTR mode. **AES-CTR Monte-Carlo Test** N/A. This is not applicable as the TOE does not claim or use AES in CTR mode. **AES Key Wrap (AES-KW) and Key Wrap with Padding (AES-KWP) Test** N/A. This is not applicable as the TOE does not claim or use AES in KW/KWP mode. **AES-CCM Tests** N/A. This is not applicable as the TOE does not claim or use AES in CCM mode. **AES-GCMP Test** N/A. This is not applicable as the TOE does not claim or use AES-GCMP mode. **AES-GCMP Monte Carlo Tests** N/A. This is not applicable as the TOE does not claim or use AES-GCMP Monte Carlo Tests. **AES-CCM Tests** N/A. This is not applicable as the TOE does not claim or use AES-CCM mode. **AES-GCM Tests** Pass. The OSPPv4.3 does not include tests for AES-GCM; however, CAVP certificate #5342 verifies that the TOE correctly implements AES-GCM.

9.2.8 FCS_COP.1/HASH

Item	Data

Test Assurance Activity	The evaluator will 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. The TSF hashing functions can be implemented in one of two modes. The first mode is the byteoriented mode. In this mode the TSF only hashes 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 test MACs. The evaluator will 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 17: Short Messages Test (Bit oriented Mode) - The evaluator will generate 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 will be pseudorandomly generated. The evaluator will 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 18: Short Messages Test (Byte oriented Mode) - The evaluator will generate 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 will be pseudorandomly generated. The evaluator will 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 19: Selected Long Messages Test (Bit oriented Mode) - The evaluator will generate 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 will be pseudorandomly generated. The evaluator will 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 20: Selected Long Messages Test (Byte oriented Mode) - The evaluator will generate 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 will be pseudorandomly generated. The evaluator will 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 21: Pseudorandomly Generated Messages Test - This test is for byte-oriented implementations only. The evaluator will 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 evaluator will then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluator will 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 #A5342 for OpenSSL for being able to implement SHA-256 (FIPS Pub 180-4), SHA-384 (FIPS Pub 180-4) and SHA2-512 (FIPS Pub 180-4). Also, the TOE was awarded the CAVP certificate #A5343 for Kernel Crypto API for being able to implement 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 #A5342 for OpenSSL for being able to implement SHA-256 (FIPS Pub 180-4), SHA-384 (FIPS Pub 180-4) and SHA2-512 (FIPS Pub 180-4). Also, the TOE was awarded the CAVP certificate #A5343 for Kernel Crypto API for being able to implement SHA2-512 (FIPS Pub 180-4). This certificate provides assurance that the TSF performs these functions as required.

9.2.9 FCS_COP.1/SIGN

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. The following content should be included if: • ECDSA schemes is selected from FCS_COP.1.1/SIGN ECDSA Algorithm Tests • Test 22: ECDSA FIPS 186-4 Signature Generation Test. For each supported NIST curve (i.e., P-384 and P-521) and SHA function pair, the evaluator will 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 will use the signature verification function of a known good implementation. • Test 23: ECDSA FIPS 186-4 Signature Verification Test. For each supported NIST curve (i.e., P-384 and P-521) and SHA function pair, the evaluator will 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 will verify that 5 responses indicate success and 5 responses indicate failure. The following content should be included if: • RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4 is selected from FCS_COP.1.1/SIGN RSA Signature Algorithm Tests

	• Test 24: Signature Generation Test. The evaluator will verify the implementation of RSA Signature Generation by the OS 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 will have the OS use its private key and modulus value to sign these messages. The evaluator will verify the correctness of the TSF' signature using a known good implementation and the associated public keys to verify the signatures. • Test 25: Signature Verification Test. The evaluator will perform the Signature Verification test to verify the ability of the OS to recognize another party's valid and invalid signatures. The evaluator will 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 evaluator will verify that the OS returns failure when validating each signature.
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 #A5342 for ECDSA signature generation and signature verification (FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5) using "NIST curves" P-384. 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 #A5342 for RSA signature generation and signature verification (FIPS Pub 186-4,"Digital Signature Standard (DSS)", Section 4) using 2048, 3072 and 4096 bit 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 #A5342 for ECDSA signature generation and signature verification (FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5) using "NIST curves" P-384. 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 #A5342 for RSA signature generation and signature verification (FIPS Pub 186-4,"Digital Signature Standard (DSS)", Section 4) using 2048, 3072 and 4096 bit RSA keys. This certificate provides assurance that the TSF performs these functions as required.

9.2.10 FCS_COP.1/KEYHMAC

Item	Data

Test Assurance Activity	The evaluator will perform the following activities based on the selections in the ST. For each of the supported parameter sets, the evaluator will compose 15 sets of test data. Each set consists of a key and message data. The evaluator will have the OS generate HMAC tags for these sets of test data. The resulting MAC tags will be compared against the result of generating HMAC tags with the same key using a known-good implementation.
Test Steps	The evaluator examined the ST and found that in "Cryptographic Support" section that the TOE was awarded the CAVP certificate #A5342 for OpenSSL library for its implementation of HMAC-SHA2-256, HMAC-SHA-384 and HMAC-SHA512 in compliance with FIPS Pub 198-1 (The Keyed-Hash Message Authentication Code) and FIPS Pub 180-4 (Secure Hash Standard). Also, the TOE was awarded the CAVP certificate #A5343 for Linux Kernel Crypto API library for its implementation of HMAC-SHA512 in compliance with FIPS Pub 198-1 (The Keyed-Hash Message Authentication Code) and FIPS Pub 180-4 (Secure Hash Standard). 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 "Cryptographic Support" section the TOE was awarded the CAVP certificate #A5342 for OpenSSL library for its implementation of HMAC-SHA2-256, HMAC-SHA-384 and HMAC-SHA512 in compliance with FIPS Pub 198-1 (The Keyed-Hash Message Authentication Code) and FIPS Pub 180-4 (Secure Hash Standard). Also, the TOE was awarded the CAVP certificate #A5343 for Linux Kernel Crypto API library for its implementation of HMAC-SHA512 in compliance with FIPS Pub 198-1 (The Keyed-Hash Message Authentication Code) and FIPS Pub 180-4 (Secure Hash Standard). This certificate provides assurance that the TSF performs these functions as required.

9.2.11 FCS_RBG_EXT.1/KERN

Item	Data
Test Assurance Activity	The evaluator will perform the following tests: The evaluator will perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator will perform 15 trials for each configuration. The evaluator will 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 will 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 will 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 list contains 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 than 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 will 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. Documentation will be produced - and the evaluator will perform the activities - in accordance with Appendix E - Entropy Documentation and Assessment and the Clarification to the Entropy Documentation and Assessment Annex. In the future, specific statistical testing (in line with NIST SP 800-90B) will be required to verify the entropy estimates.
Test Steps	**Implementations Conforming to NIST SP 800-57.** This is not applicable as the TOE does not claim implementations conforming to NIST SP 800-57. **Implementations Conforming to NIST Special Publication 800-90A.** The evaluator examined the ST and found that in Section 6.2 that the TOE was awarded the CAVP certificate #A5343 for HMAC_DRBG (SHA2-512). This certificate provides assurance that the TSF performs these functions as required.
Pass/Fail with Explanation	**Implementations Conforming to NIST SP 800-57.** N/A. This is not applicable as the TOE does not claim implementations conforming to NIST SP 800-57. **Implementations Conforming to NIST Special Publication 800-90A.** Pass. The evaluator examined the ST and found that in Section 6.2 that the TOE was awarded the CAVP certificate #A5343 for HMAC_DRBG (SHA2-512).

	This certificate provides assurance that the TSF performs these functions as required.

9.2.12 FCS_RBG_EXT.1/OSSL

Item	Data
Test Assurance Activity	The evaluator will perform the following tests: The evaluator will perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator will perform 15 trials for each configuration. The evaluator will 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 will 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 will 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 list contains 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 than 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 will 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.

	Documentation will be produced - and the evaluator will perform the activities - in accordance with Appendix E - Entropy Documentation and Assessment and the Clarification to the Entropy Documentation and Assessment Annex. In the future, specific statistical testing (in line with NIST SP 800-90B) will be required to verify the entropy estimates.
Test Steps	**Implementations Conforming to NIST SP 800-57.** This is not applicable as the TOE does not claim implementations conforming to NIST SP 800-57. **Implementations Conforming to NIST Special Publication 800-90A.** The evaluator examined the ST and found that in Section 6.2 that the TOE was awarded the CAVP certificate #A5342 for CTR_DRBG (AES-256). This certificate provides assurance that the TSF performs these functions as required.
Pass/Fail with Explanation	**Implementations Conforming to NIST SP 800-57.** N/A. This is not applicable as the TOE does not claim implementations conforming to NIST SP 800-57. **Implementations Conforming to NIST Special Publication 800-90A.** Pass. The evaluator examined the ST and found that in Section 6.2 that the TOE was awarded the CAVP certificate #A5342 for CTR_DRBG (AES-256). This certificate provides assurance that the TSF performs these functions as required.

9.2.13 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	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384: 1. Start a TLS server that supports the cipher suite:"TLS_DHE_RSA_WITH_AES_256_GCM_SHA384". 2. Attempt to connect to the TLS server from the TOE. 3. Verify that the correct cipher suite was used. 4. Verify that the connection succeeds via packet capture. TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384: 1. Start a TLS server that supports the cipher suite TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384. 2. Attempt to connect to the TLS server from the TOE. 3. Verify that the correct cipher suite was used. 4. Verify that the connection succeeds via packet capture.

	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384: 1. Start a TLS server that supports the cipher suite TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384. 2. Attempt to connect to the TLS server from the TOE. 3. Verify that the correct cipher suite was used. 4. Verify that the connection succeeds via packet capture.
Expected Test Results	The TOE should connect to a TLS server using each of the cipher suites selected in the ST.
Pass/Fail with Explanation	Pass. The TOE successfully connects to the TLS server using each of the cipher suites selected in the ST. This satisfies the testing requirement.

9.2.14 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	1. Start a TLS server on the test VM using a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field. 2. Attempt to connect to the TLS server from the TOE. 3. Verify that the connection was successful using packet capture 4. Start a TLS server on the test VM using a server certificate that does not contain the Server Authentication purpose in the extendedKeyUsage field. 5. Attempt to connect to the TLS server from the TOE. 6. Verify that the connection failed using packet capture.
Expected Test Results	The TOE should connect to a TLS server that is using a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and not connect to a TLS server that does not contain the Server Authentication purpose in the extendedKeyUsage field.
Pass/Fail with Explanation	Pass. The TOE connects to the TLS server that is using a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and does not connect to the TLS server that does not contain the Server Authentication purpose in the extendedKeyUsage field.

9.2.15 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	1. Using the acumen tlsc tool, start a tls server that will send an EC server certificate that does not match the server-selected cipher suite. 2. Attempt to connect to the TLS server from the TOE. 3. Verify that the connection fails via packet capture.
Expected Test Results	The TOE should disconnect from the TLS server when the server certificate does not match the server-selected cipher.
Pass/Fail with Explanation	Pass. The TOE disconnects from the TLS server when the server certificate does not match the server-selected cipher. This satisfies the testing requirement.

9.2.16 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	1. The evaluator configures the server to use the cipher suite TLS_NULL_WITH_NULL_NULL. 2. Starts a TLS connection between the TOE and the TLS server. 3. Verify connection fails via packet capture
Expected Test Results	The TOE should deny a connection to a TLS server that is using the TLS_NULL_WITH_NULL_NULL cipher suite.
Pass/Fail with Explanation	Pass. The TOE denies a connection to a TLS server that is using the TLS_NULL_WITH_NULL_NULL cipher suite. This satisfies the testing requirement.

9.2.17 FCS_TLSC_EXT.1.1 Test#5.1

Item	Data
Test Assurance Activity	The evaluator shall perform the following modifications to the traffic: 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	1. Start a TLS server that will send an undefined tls version. 2. On the TOE attempt a TLS connection to the "Acumen-TLSC" server and verify the connection fails. 3. Verify the unsuccessful connection via packet capture.
Expected Test Results	The TOE should reject a connection to a TLS server that sends an undefined TLS version.
Pass/Fail with Explanation	Pass. The TOE rejects a connection to a TLS server that has an undefined version of TLS. This satisfies the testing requirement.

9.2.18 FCS_TLSC_EXT.1.1 Test#5.2

Item	Data
Test Assurance Activity	The evaluator shall perform the following modifications to the traffic: 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	Note: the TOE only supports TLS version 1.2 1. Configure the server to use the most recent unsupported TLS version(TLS version 1.1). 2. Attempt to connect to the TLS server from the TOE. 3. Verify that the connection failed using packet capture.
Expected Test Results	The TOE should reject a connection to a TLS server that sends the most recent unsupported TLS version.
Pass/Fail with Explanation	Pass. The TOE rejects a connection to a TLS server that sends the most recent unsupported TLS version (TLSv1.1). This satisfies the testing requirement.

9.2.19 FCS_TLSC_EXT.1.1 Test#5.3

Item	Data
Test Assurance Activity	The evaluator shall perform the following modifications to the traffic: [conditional] If **DHE or ECDHE cipher suites are supported**, modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the client does not complete the handshake and no application data flows.
Test Steps	1. Start the acumen-tlsc tool so that it will modify at least one byte in the server's nonce in the Server Hello handshake message. 2. Attempt to connect to the acumen-tlsc tool from the TOE. 3. Verify that the connection fails via packet capture.
Expected Test Results	The TOE should deny a connection when at least one byte in the server's nonce in the Server Hello handshake message is modified.
Pass/Fail with Explanation	Pass. The TOE denies connection when at least one byte in the server's nonce in the Server Hello handshake message is modified. This satisfies the testing requirement.

9.2.20 FCS_TLSC_EXT.1.1 Test#5.4

Item	Data
Test Assurance Activity	The evaluator shall perform the following modifications to the traffic: 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	1. Start the acumen-tlsc tool so that it will modify selected cipher suite. 2. Attempt to connect to the acumen-tlsc tool from the TOE. 3. Verify that the connection fails via packet capture.
Expected Test Results	The TOE should deny a connection to a TLS server when the server's selected cipher suite in the Server Hello handshake message is not present in the Client Hello handshake message.

Pass/Fail with Explanation	Pass. The TOE denies connection to a TLS server when the server's selected cipher suite in the Server Hello handshake message does not match with a cipher present in the Client Hello handshake message. This satisfies the testing requirement.

9.2.21 FCS_TLSC_EXT.1.1 Test#5.5

Item	Data
Test Assurance Activity	The evaluator shall perform the following modifications to the traffic: [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	1. Start the acumen-tlsc tool so that it will modify the signature block in the server's Key exchange handshake message. 2. Attempt to connect to the acumen-tlsc tool from the TOE. 3. Verify that the connection fails and no application data flows via packet capture.
Expected Test Results	The TOE should deny a connection to a TLS server when the signature block in the Server's Key Exchange handshake message is modified.
Pass/Fail with Explanation	Pass. The TOE denies connection to a TLS server when the signature block in the Server's Key Exchange handshake message is modified. This satisfies the testing requirement.

9.2.22 FCS_TLSC_EXT.1.1 Test#5.6

Item	Data
Test Assurance Activity	The evaluator shall perform the following modifications to the traffic: 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	1. Start the acumen-tlsc tool so that it will modify a byte in Server Finished handshake 2. Attempt to connect to the acumen-tlsc tool from the TOE 3. Verify that the handshake is not complete and no application data flows via packet capture
Expected Test Results	The TOE should deny a connection to a TLS server when a byte in the Server Finished handshake message is modified.
Pass/Fail with Explanation	Pass. The TOE denies connection to a TLS server when a byte in the Server Finished handshake message is modified. This satisfies the testing requirement.

9.2.23 FCS_TLSC_EXT.1.1 Test#5.7

Item	Data
Test Assurance Activity	The evaluator shall perform the following modifications to the traffic: 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	1. Start the acumen-tlsc tool and configure it to send a message consisting of random bytes after sending the Change Cipher spec message. 2. Attempt to connect to the acumen-tlsc tool from the TOE 3. Verify the handshake is not complete and the message still have a valid 5-byte record header using packet capture.
Expected Test Results	The TOE should not complete the handshake when message with random bytes messages is sent.
Pass/Fail with Explanation	Pass. The TOE does not complete the handshake when message with random bytes is transmitted after the ChangeCipherSpec message is sent. This satisfies the testing requirement.

9.2.24 FCS_TLSC_EXT.1.2 Test#1 (TD0499)

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.
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. The ST states the following: “If the SAN is not present, the referenced identifier is matched against the CN for DNS.” Using an FQDN as a reference identifier in the CN field: 1. The evaluator started a TLS server that used an X509 certificate that contains an FQDN (wrong.hname) in the CN field that does not match the reference identifier and does not contain a SAN extension. 2. The evaluator attempted to establish a TLS connection with the TOE using the reference identifier idtech.toe and the connection was unsuccessful. 3. Verify that the connection fails using a packet capture.
Expected Test Results	The TOE should deny a connection to a TLS server when the server certificate does not contain and identifier in either the SAN extension or CN field that matched the reference identifier.
Pass/Fail with Explanation	Pass. The TOE denies a connection when a server certificate with an FQDN in the CN field does not match the reference identifier and does not contain a SAN extension. This satisfies the testing requirements.

9.2.25 FCS_TLSC_EXT.1.2 Test#2 (TD0499)

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.
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Using an IP address as a reference identifier: 1. The evaluator started a TLS server on the test VM with a certificate that contains an IP address in the CN field that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. 2. The evaluator then attempted to connect to the server from the TOE. 3. The evaluator then verified that the connection failed using packet capture. Using an FQDN as a reference identifier: 4. The evaluator started a TLS server on the test VM with a certificate that contains an FQDN in the CN field that matches the reference identifier, contains the SAN extension (FQDN), but does not contain an identifier in the SAN that matches the reference identifier. 5. The evaluator then attempted to connect to the server from the TOE. 6. The evaluator then verified that the connection failed using packet capture.
Expected Test Results	The TOE should deny a connection to a TLS server when the server certificate contains an FQDN or an IP address in the CN field that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier.
Pass/Fail with Explanation	Pass. The TOE denies connection to a TLS server when the server certificate contains an FQDN or an IP address in the CN field that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. This satisfies the testing requirement.

9.2.26 FCS_TLSC_EXT.1.2 Test#3 (TD0499)

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.
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. *The ST states the following :”If the SAN is not present, the referenced identifier is matched against the CN for DNS. For IP address, the TOE matches the reference identifier against the SAN only.*” Using an FQDN as a reference identifier: 1. The evaluator started a TLS server on the test VM that uses a certificate that contains a CN (DNS) that matches the reference identifier and does not contain the SAN extension. 2. The evaluator then attempted to connect to the TLS server from the TOE. 3. Verify the connection succeeds using packet capture.
Expected Test Results	The TOE should connect to a TLS server with a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension.
Pass/Fail with Explanation	Pass. The TOE accepts a connection when a server certificate with an FQDN in the CN field matches the reference identifier and does not contain a SAN extension. This satisfies the testing requirements.

9.2.27 FCS_TLSC_EXT.1.2 Test#4 (TD0499)

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.
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. The ST states the following :” For IP address, the TOE matches the reference identifier against the SAN only.”, also “The TOE establishes the reference

	identifier by parsing the DNS Name or IP address for the configured TLS server.” Using an IP address as a reference identifier: 1. The evaluator started a TLS server on the test VM with a certificate that contains an IP address in the CN field that does not match the reference identifier but does contain an IP address in the SAN extension that matches. 2. The evaluator then attempted to connect to the TLS server from the TOE. 3. The evaluator then verified that the connection was successful using packet capture. Using an FQDN as a reference identifier: 4. The evaluator started a TLS server on the test VM with a certificate that contains a DNS name in the CN field that does not match the reference identifier but does contain a DNS name identifier in the SAN extension that matches. 5. The evaluator then attempted to connect to the TLS server from the TOE. 6. The evaluator then verified that the connection was successful using packet capture.
Expected Test Results	The TOE should connect to a TLS server with a server certificate that contains an FQDN or an IP address in the CN field that does not match the reference identifier but does contain an identifier in the SAN that matches the reference identifier.
Pass/Fail with Explanation	Pass. The TOE connects to a TLS server with a server certificate that contains an IP address or FQDN in the CN field that does not match the reference identifier but does contain an identifier in the SAN extension that matches the reference identifier. This satisfies the testing requirement.

9.2.28 FCS_TLSC_EXT.1.2 Test#5.1 (TD0499)

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.

Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Testing wildcard (e.g. idtech.*.archon.toe) in the SAN extension: 1. The evaluator started a TLS server on the test VM with a certificate containing a wildcard in the SAN extension that is not in the left-most label of the presented identifier (e.g. idtech.*.archon.toe) in the SAN extension. 2. The evaluator then attempted to connect to the server from the TOE, using the reference identifier: idtech.acumen.archon.toe. 3. The evaluator then verified that the connection fails. Testing wildcard (e.g. idtech.*.archon.toe) in the CN field:..The evaluator started a TLS server on the test VM with a certificate containing a wildcard in the CN field that is not in the left-most label of the presented identifier (e.g. idtech.*.archon.toe) in the CN field..The evaluator then attempted to connect to the server from the TOE, using the reference identifier: idtech.acumen.archon.toe..The evaluator then verified that the connection fails.
Expected Test Results	The TOE should not connect to a TLS server with a server a certificate containing a wildcard in the SAN extension or in CN field that is not in the left-most label.
Pass/Fail with Explanation	Pass. The TOE does not connect to a TLS server with a server a certificate containing a wildcard in the SAN extension or CN field that is not in the left-most label (e.g., foo.*.example.com). This satisfies the testing requirement.

9.2.29 FCS_TLSC_EXT.1.2 Test#5.2 (TD0499)

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. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails.

	The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.come) and verify that the connection fails.
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Testing wildcard in the SAN extension: 1. The evaluator created an x509 certificate containing a wildcard in the SAN extension, in the left-most label but not preceding the public suffix (*.idtech.toe) and started a TLS server on the test VM using the newly created x509 certificate. 2. The evaluator then attempted to connect to the TLS server from the TOE using the reference identifier acumen.idtech.toe and was successful. 3. Using network traffic capture, the evaluator verified that the connection succeeded. 4. The evaluator then attempted to connect to the TLS server from the TOE using the reference identifier idtech.toe and was unsuccessful. 5. Using network traffic capture, the evaluator verified that the connection failed. 6. The evaluator then attempted to connect to the TLS server from the TOE using the reference identifier cctest.acumen.idtech.toe and was unsuccessful. 7. Using network traffic capture, the evaluator verified that the connection failed. Testing wildcard in the CN field: 8. The evaluator created an x509 certificate containing a wildcard in the CN field, in the left-most label but not preceding the public suffix (*.idtech.toe) and started a TLS server on the test VM using the newly created x509 certificate. 9. The evaluator then attempted to connect to the TLS server from the TOE using the reference identifier acumen.idtech.toe and was successful. 10. Using network traffic capture, the evaluator verified that the connection succeeded. 11. The evaluator then attempted to connect to the TLS server from the TOE using the reference identifier idtech.toe and was unsuccessful. 12. Using network traffic capture, the evaluator verified that the connection failed. 13. The evaluator then attempted to connect to the TLS server from the TOE using the reference identifier cctest.acumen.idtech.toe and was unsuccessful. 14. Using network traffic capture, the evaluator verified that the connection failed.

Expected Test Results	The TOE should connect to a TLS server that is using a server certificate containing a wildcard in the left-most label but not preceding the public suffix (e.g. *.idtech.toe) when the configured reference identifier is (e.g. acumen.idtech.toe) but not connect when the configured reference identifier is (e.g. idtech.toe, and e.g. cctest.acumen.idtech.toe).
Pass/Fail with Explanation	Pass. The TOE connects to a TLS server that is using a server certificate containing a wildcard in SAN extension or CN field, in the left-most label but not preceding the public suffix (e.g. *.itech.toe) when the configured reference identifier is (e.g. acumen.idtech.toe) but does not connect when the configured reference identifier is (idtech.toe, and cctest.acumen.idtech.toe). This satisfies the testing requirement.

9.2.30 FCS_TLSC_EXT.1.2 Test#5.3 (TD0499)

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. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.com) and verify that the connection fails.
Test Steps	Note: The TOE does not support certificate pinning; therefore, no pinned certificate needs to be removed before performing tests 1 through 6. Tested wildcard (*.toe) in the SAN extension: using the reference Identifier (idtech.toe) 1. The evaluator started a TLS server on the test VM with a certificate containing the wildcard: *.toe in SAN extension, in the left-most label immediately preceding the public suffix. 2. The evaluator then attempted to connect to the TLS server from the TOE, using the reference identifier: idtech.toe. 3. The evaluator then verified that the connection failed using packet capture Using the reference Identifier (archon.idtech.toe)

	4. The evaluator then attempted to connect to the TLS server from the TOE, using the reference identifier: archon.idtech.toe. 5. The evaluator then verified that the connection failed using packet capture. Tested Wildcard in the CN field: Using the reference Identifier (idtech.toe) 1. The evaluator then attempted to connect to the TLS server from the TOE, using the reference identifier: idtech.toe. 2. The evaluator then verified that the connection failed using packet capture. Using the reference Identifier (archon.idtech.toe) 3. The evaluator then attempted to connect to the TLS server from the TOE, using the reference identifier: archon.idtech.toe. 4. The evaluator then verified that the connection failed using packet capture
Expected Test Results	The TOE should not connect to a TLS server that is using a server certificate containing a wildcard in the SAN extension or in the CN field, in the left-most label immediately preceding the public suffix (e.g. *.toe) when the configured reference identifier is (e.g. idtech.toe) or (e.g. archon.idtech.toe).
Pass/Fail with Explanation	Pass. The TOE does not connect to a TLS server that is using a server certificate containing a wildcard in the SAN extension or CN field, in the left-most label immediately preceding the public suffix (e.g. *.toe) when the configured reference identifier is (e.g. idtech.toe) or (e.g. archon.idtech.toe).This satisfies the testing requirement.

9.2.31 FCS_TLSC_EXT.1.2 Test#5.4 (TD0499)

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 not supported, the evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection fails.
Pass/Fail with Explanation	N/A. As the TOE does not support certificate pinning; and wildcards are supported by the TOE.

9.2.32 FCS_TLSC_EXT.1.2 Test#6 (TD0499)

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.
Pass/Fail with Explanation	N/A, as the TOE does not support certificate pinning; and the TOE does not support URI or Service name reference identifiers.

9.2.33 FCS_TLSC_EXT.1.2 Test#7 (TD0499)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection. If the TOE supports certificate pinning, all pinned certificates must be removed before performing Tests 1 through 6. A pinned certificate must be added prior to performing Test 7. Test 7: [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, as pinned certificates are not supported.

9.2.34 FCS_TLSC_EXT.1.3 Test#1a (TD0513)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that a server using a certificate with a valid certification path successfully connects.
Test Steps	1. The evaluator configured the TOE and the update server to use appropriate certificates. 2. The evaluator initiated an update check on the TOE. 3. The evaluator verified that the TOE successfully connects to the update server over TLS.
Expected Test Results	The TOE should successfully connect to an update server when it presents a valid x509 certification path.
Pass/Fail with Explanation	Pass. The TOE successfully connects to a server with a valid certification path. This satisfies the testing requirement.

9.2.35 FCS_TLSC_EXT.1.3 Test#1b (TD0513)

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.
Test Steps	1. The evaluator modified the certificate chain on the update server. 2. The evaluator initiated a software update check on the TOE. 3. The evaluator verified that the check for software update failed via packet capture.
Expected Test Results	The TOE should not connect to an update server when it presents an invalid x509 certification path.
Pass/Fail with Explanation	Pass. The TOE does not connect to an update server when it presents an invalid x509 certification path. This satisfies the testing requirement.

9.2.36 FCS_TLSC_EXT.1.3 Test#1c (TD0513)

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.
Test Steps	1. The evaluator deleted ICA2 certificate stored in the trust store of the TOE and ran the command 'update-ca-trust' to update it. 2. The evaluator initiated a software update check on the TOE after ICA2 certificate was deleted. 3. Using packet capture, the evaluator verified that the TOE connection to the update server failed.
Expected Test Results	The TOE should not authenticate to an update server that presents untrusted X509 certificate chain.
Pass/Fail with Explanation	Pass. The TOE does not authenticate to an update server that presents an untrusted X509 certificate chain. This satisfies the testing environment.

9.2.37 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. Satisfied by FIA_X509_EXT.1.1 Test 66 testing assurance activity.

9.2.38 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.
Test Steps	1. The evaluator established a TLS session between the TOE and the TLS server. 2. The evaluator provided a certificate that was past its expiration date when the TLS connection was attempted. 3. The evaluator verified using packet capture that the TLS connection was not established.
Expected Test Results	The TOE should not establish a TLS connection when an expired x509 certificate is presented.
Pass/Fail with Explanation	Pass. The TOE does not establish a TLS connection when an expired x509 certificate is presented. This satisfies the testing requirement.

9.2.39 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	1. The evaluator established a TLS session between the TOE and the TLS server. 2. The evaluator provided a certificate that does not have a valid identifier when connection is attempted. 3. The evaluator verified that the connection is not established using packet capture.
Expected Test Results	The TOE should not establish a session with a certificate without valid identifiers.
Pass/Fail with Explanation	Pass. The TOE does not establish a session with a certificate without valid identifiers. This satisfies the testing requirement.

9.2.40 FCS_TLSC_EXT.2.1 Test#1

Item	**Data**
Test Assurance Activity	The evaluator shall also perform the following tests: • Test 1: The evaluator shall establish a connection to a server that is not configured for mutual authentication (i.e. does not send Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE did not send Client's Certificate message (type 11) during handshake.
Test Steps	Application based: 1. Verify the server and client is not configured for mutual authentication. 2. Establish a connection between server and TOE. 3. Verify the Server's Certificate request (type 13) message is not present using packet capture. 4. Verify the TOE did not respond with a non-empty Client's Certificate message (type 11) during handshake using packet capture.

	Update Server: 1. Verify the update server and TOE is not configured for mutual authentication. 2. Initiate an update from the TOE. 3. Verify the Server's Certificate request (type 13) message is not present using packet capture. 4. Verify the TOE did not respond with a non-empty Client's Certificate message (type 11) during handshake using packet capture.
Expected Test Results	The TOE should not return a Client's Certificate message when the Server's Certificate Request is not presented by a TLS server.
Pass/Fail with Explanation	Pass. The TOE does not return a Client's Certificate message when the Server's Certificate Request is not presented. This satisfies the testing requirement.

9.2.41 FCS_TLSC_EXT.2.1 Test#2

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests: • Test 2: The evaluator shall establish a connection to a server with a shared trusted root that is configured for mutual authentication (i.e. it sends Server's Certificate Request (type 13) message). The evaluator observes negotiation of a TLS channel and confirms that the TOE responds with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message.
Test Steps	Application based: 1. Verify the server and client have a shared trusted root configured. 2. Initiate connection between the server and the TOE. 3. Verify the connection succeeded. 4. Verify the Server's Certificate request (type 13) message using packet capture. 5. Verify the TOE responded with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message using packet capture. Update Server 1. Verify the server and client have a shared trusted root configured. 2. Initiate an update on the TOE. 3. Verify the connection succeeded. 4. Verify the Server's Certificate request (type 13) message using packet capture. 5. Verify the TOE responded with a non-empty Client's Certificate message (type 11) and Certificate Verify (type 15) message using packet capture.
Expected Test Results	The TOE is to return a Client's Certificate Message (Type 11 and Type 15) when Server Certificate Request (Type 13) is presented.
Pass/Fail with Explanation	Pass. The TOE returns a Client's Certificate Message (Type 11 and type 15) when Server Certificate Request (Type 13) is presented. This satisfies the testing requirement.

9.2.42 FCS_TLSC_EXT.4.1 Test#1

Item	Data

Test Assurance Activity	The evaluator shall perform the following tests: The evaluator shall use a network packet analyzer/sniffer to capture the traffic between the two TLS endpoints. The evaluator shall verify that either the "renegotiation_info" field or the SCSV cipher suite is included in the ClientHello message during the initial handshake.
Test Steps	1. Attempt a TLS connection from the TOE to the TLS Server. 2. Verify the SCSV Cipher Suite in the packet capture.
Expected Test Results	Packet capture evidence showing SCSV cipher suite included in the TOE's ClientHello message.
Pass/Fail with Explanation	Pass. The SCSV cipher suite is included in the TOE's ClientHello message during the initial handshake. This satisfies the testing requirements.

9.2.43 FCS_TLSC_EXT.4.1 Test#2

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: The evaluator shall verify the Client's handling of ServerHello messages received during the initial handshake that include the "renegotiation_info" extension. The evaluator shall modify the length portion of this field in the ServerHello message to be non-zero and verify that the client sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.
Test Steps	1. Attempt a TLS connection using "Acumen-TLSC" tool to modify the length portion of the "renegotiation_info" field in the ServerHello message to be non-zero and verify that the connection fails. 2. Verify the unsuccessful connection with the packet capture. 3. Attempt a TLS connection using "Acumen-TLSC" tool with a properly formatted field without any modification. 4. Verify the successful connection with the packet capture.
Expected Test Results	• Packet capture evidence showing unsuccessful connection when the length portion of the "renegotiation_info" field in the ServerHello message is modified to be non-zero. • Packet capture evidence showing successful connection with a properly formatted field without any modification.
Pass/Fail with Explanation	Pass. The TOE terminates the connection when the length portion of the "renegotiation_info" field in the ServerHello message is modified to be non-zero and accepts the connection with a properly formatted field. This satisfies the testing requirement.

9.2.44 FCS_TLSC_EXT.4.1 Test#3

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: The evaluator shall verify that ServerHello messages received during secure renegotiation contain the "renegotiation_info" extension. The evaluator shall modify either the "client_verify_data" or "server_verify_data" value and verify that the client terminates the connection.
Test Steps	1. Attempt a TLS secure renegotiation connection. 2. Verify that the ServerHello messages received during secure renegotiation contain the "renegotiation_info" extension.

	3. Attempt to modify the “sever_verify_data” value and verify that the connection fails. 4. Verify with packet capture that the ServerHello contains the “renegotiation_info” extension. 5. Using a decrypted packet capture. Verify the TOE’s unsuccessful connection after modification of the TLS server’s “server_verify_data” field.
Expected Test Results	The TOE will terminate a TLS connection when the "server_verify_data" value in the "renegotiation_info" extension in a ServerHello message is modified.
Pass/Fail with Explanation	Pass. The TOE terminates a TLS connection when the "server_verify_data" value in the "renegotiation_info" extension in a ServerHello message is modified. This satisfies the testing requirements.

9.2.45 FCS_TLSC_EXT.5.1 Test#1

Item	Data
Test Assurance Activity	The evaluator shall also perform the following test: • **Test 1:** The evaluator shall configure a server to perform key exchange using each of the TOE’s supported curves and/or groups. The evaluator shall verify that the TOE successfully connects to the server.
Test Steps	1. Start a TLS server using a ECDSA certificate and configure it to use the curve [secp384r1] in its server key exchange message. 2. Attempt to connect to the TLS server from the TOE. 3. Verify that the certificate is using the claimed curve. 4. Verify that the connection succeeds using packet capture.
Expected Test Results	The TOE should successfully establish a connection when selected curve is presented in the certificate.
Pass/Fail with Explanation	Pass. The TOE successfully establishes a connection when the selected curve is presented in the certificate. This satisfies the testing requirement.

9.3 FDP

9.3.1 FDP_ACF_EXT.1 Test#26

Item	Data
Test Assurance Activity	The evaluator will create two new standard user accounts on the system and conduct the following tests: • **Test 26:** The evaluator will authenticate to the system as the first user and create a file within that user's home directory. The evaluator will then log off the system and log in as the second user. The evaluator will then attempt to read the file created in the first user's home directory. The evaluator will ensure that the read attempt is denied.
Test Steps	1. Create two new standard users on the TOE. 2. Log into the TOE as the User A (test_1). 3. Create a file in the home directory. 4. Verify the audit logs that file was created. 5. Log into the TOE as User B (test_2).

	6. Attempt to read the file created by User A(test_1). 7. Verify the audit log that the attempt fails.
Expected Test Results	A user should not be able to read a file in another user's home directory.
Pass/Fail with Explanation	Pass. The TOE does not allow a user to read files in another user's home directory. This satisfies the testing requirement.

9.3.2 FDP_ACF_EXT.1 Test#27

Item	Data
Test Assurance Activity	The evaluator will create two new standard user accounts on the system and conduct the following tests: - **Test 27:** The evaluator will authenticate to the system as the first user and create a file within that user's home directory. The evaluator will then log off the system and log in as the second user. The evaluator will then attempt to modify the file created in the first user's home directory. The evaluator will ensure that the modification is denied.
Test Steps	1. Log into the TOE as the first user(test_1). 2. Create a file in the home directory. 3. Log into the TOE as a second user(test_2). 4. Attempt to modify the file. 5. Using the audit logs, verify that the attempt fails.
Expected Test Results	A user should not be able to modify a file in another user's home directory.
Pass/Fail with Explanation	Pass. The TOE does not allow a user to modify a file in another user's home directory. This satisfies the testing requirement.

9.3.3 FDP_ACF_EXT.1 Test#28

Item	Data
Test Assurance Activity	The evaluator will create two new standard user accounts on the system and conduct the following tests: - **Test 28:** The evaluator will authenticate to the system as the first user and create a file within that user's user directory. The evaluator will then log off the system and log in as the second user. The evaluator will then attempt to delete the file created in the first user's home directory. The evaluator will ensure that the deletion is denied.
Test Steps	1. Log into the TOE as the first user (test_1). 2. Create a file in the home directory. 3. Log into the TOE as a second user (test_2). 4. Attempt to delete the file. 5. Using the audit logs, verify that the attempt fails.
Expected Test Results	A user should not be able to delete a file in another user's home directory.
Pass/Fail with Explanation	Pass. The TOE does not allow a user to delete a file in another user's home directory. This satisfies the testing requirement.

9.3.4 FDP_ACF_EXT.1 Test#29

Item	Data

Test Assurance Activity	The evaluator will create two new standard user accounts on the system and conduct the following tests: • **Test 29:** The evaluator will authenticate to the system as the first user. The evaluator will attempt to create a file in the second user's home directory. The evaluator will ensure that the creation of the file is denied.
Test Steps	1. Log into the TOE as the first user (test_1). 2. Attempt to create a file in the second user’s (test_2) home directory. 3. Using the audit logs verify that it fails.
Expected Test Results	User is not allowed to create a file in another user's home directory
Pass/Fail with Explanation	Pass. The TOE does not allow a user to create a file in another user’s home directory. This satisfies the testing requirement.

9.3.5 FDP_ACF_EXT.1 Test#30

Item	Data
Test Assurance Activity	The evaluator will create two new standard user accounts on the system and conduct the following tests: • Test 30: The evaluator will authenticate to the system as the first user and attempt to modify the file created in the first user's home directory. The evaluator will ensure that the modification of the file is accepted.
Test Steps	1. Log into the TOE as the first user (test_1). 2. Create a file in the first user’s home directory. 3. Attempt to modify that file as the first user (test_1). 4. Using the audit logs verify that the modification attempt succeeds.
Expected Test Results	A user should be able to modify a file that they created in their own home directory.
Pass/Fail with Explanation	Pass. The TOE allows a user to modify file they created in their own home directory. This satisfies the testing requirement.

9.3.6 FDP_ACF_EXT.1 Test#31

Item	Data
Test Assurance Activity	The evaluator will create two new standard user accounts on the system and conduct the following tests: • Test 31: The evaluator will authenticate to the system as the first user and attempt to delete the file created in the first user's directory. The evaluator will ensure that the deletion of the file is accepted.
Test Steps	1. Log into the TOE as user test_1. 2. Create a file in the home directory of user test_1. 3. Attempt to delete the file while logged into user test_1. 4. Using the audit logs, verify that the attempt has succeeded.
Expected Test Results	A user should be able to delete a file that he created in his own home directory.
Pass/Fail with Explanation	Pass. The TOE does allow a user to delete a file that he created in his own home directory. This satisfies the testing requirement.

9.4 FMT

9.4.1 FMT_MOF_EXT.1 Test#32

Item	Data
Test Assurance Activity	• Test 32: For each function that is indicated as restricted to the administrator, the evaluation will perform the function as an administrator, as specified in the Operational Guidance, and determine that it has the expected effect as outlined by the Operational Guidance and the SFR. The evaluator will then perform the function (or otherwise attempt to access the function) as a non-administrator and observe that they are unable to invoke that functionality.
Test Steps	Enable Session timeout. Successful - admin • Before • Attempt changes. • After • Logs Unsuccessful - user • Before • Attempt changes. • After • Logs Disable Session timeout. Successful - admin • Before • Attempt changes. • After • Logs Unsuccessful - user • Before • Attempt changes. • After • Logs Configure [session] inactivity timeout. Successful - admin • Before • Attempt changes. • After • Logs Unsuccessful - user • Before • Attempt changes.

	• After • Logs Import keys/secrets into the secure key storage. Successful - admin • Before • Attempt changes. • After • Logs Unsuccessful - user • Before • Attempt changes. • After • Logs Configure local audit storage capacity. Successful - admin • Before • Attempt changes. • After • Logs Unsuccessful - user • Before • Attempt changes. • After • Logs Configure minimum password length. Successful - admin • Before • Attempt changes. • After • Logs Unsuccessful - user • Before • Attempt change. • After • Logs Configure minimum number of special characters in password. Successful - admin • Before • Attempt changes. • After • Logs Unsuccessful - user • Before

	• Attempt changes • After • Logs Configure minimum number of numeric characters in password. Successful - admin • Before • Attempt changes. • After • Logs Unsuccessful - user • Before • Attempt changes. • After • Logs Configure minimum number of uppercase characters in password. Successful - admin • Before • Attempt changes. • After • Logs Unsuccessful - user • Before • Attempt changes. • After • Logs Configure minimum number of lowercase characters in password Successful - admin • Before • Attempt changes • After • Logs Unsuccessful - user • Before • Attempt changes • After • Logs Configure lockout policy for unsuccessful authentication attempts through [timeouts between attempts] Successful - admin • Before • Attempt changes • After

	- Logs Unsuccessful - user - Before - Attempt changes - After - Logs
Expected Test Results	The TOE should allow an admin to perform admin functions and restrict a non-admin from performing the admin functions.
Pass/Fail with Explanation	Pass. The TOE restricts configuration changes to privileged users. This satisfies the testing requirements.

9.4.2 FMT_SMF_EXT.1

Item	Data
Test Assurance Activity	The evaluator will test the OS's ability to provide the management functions by configuring the operating system and testing each option selected from above. The evaluator is expected to test these functions in all the ways in which the ST and guidance documentation state the configuration can be managed.
Pass/Fail with Explanation	Pass. The test requirements are satisfied by FMT_MOF_EXT.1.

9.5 FPT

9.5.1 FPT_ACF_EXT.1 Test#33

Item	Data
Test Assurance Activity	The evaluator will create an unprivileged user account. Using this account, the evaluator will ensure that the following tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action): - Test 33: The evaluator will attempt to modify all kernel drivers and modules.
Test Steps	Note: The TOE uses these permissions to protect the following from unauthorized modification: - Kernel, drivers, and kernel modules – files in: o /boot/ o /usr/lib/modules/ o /usr/lib/firmware/ 1. Create an unprivileged user (user_unp). 2. Verify through logs that the created user is an unprivileged user. 3. Login and navigate to the directory /boot/ . 4. Attempt to modify files as an unprivileged user. 5. Verify that the attempt failed. 6. Verify through logs that attempts at modification have failed. 7. Log into the TOE as an unprivileged user (user_unp). 8. Navigate to the directory /usr/lib/firmware 9. Attempt to modify files as an unprivileged user.

	10. Verify that the attempt failed. 11. Verify through logs that attempts at modification have failed. 12. Log into the TOE as an unprivileged user (user_unp). 13. Navigate to the directory /usr/lib/modules 14. Attempt to modify files as an unprivileged user. 15. Verify that the attempt failed. 16. Verify through logs that attempts at modification have failed.
Expected Test Results	The TOE should not allow an unprivileged user to modify the kernel drivers.
Pass/Fail with Explanation	Pass. The TOE does not allow an unprivileged user to modify the kernel drivers and modules. This satisfies the testing requirements.

9.5.2 FPT_ACF_EXT.1 Test#34

Item	Data
Test Assurance Activity	The evaluator will create an unprivileged user account. Using this account, the evaluator will ensure that the following tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action): • **Test 34:** The evaluator will attempt to modify all security audit logs generated by the logging subsystem.
Test Steps	Note: The TOE uses these permissions to protect the following from unauthorized modification: • Security audit logs – files in: o /var/log/audit/ o /var/log/ 1. Log into the TOE as an unprivileged user. 2. Navigate to the directory /var. 3. Show contents of log file prior to modification. 4. Attempt to modify all security audit logs. 5. Verify that the modifications fail via audit logs.
Expected Test Results	The TOE should not allow a non-privileged user to modify TOE logs.
Pass/Fail with Explanation	Pass. The TOE does not allow a non-privileged user to modify TOE logs. This satisfies the testing requirement.

9.5.3 FPT_ACF_EXT.1 Test#35

Item	Data
Test Assurance Activity	The evaluator will create an unprivileged user account. Using this account, the evaluator will ensure that the following tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action): • **Test 35:** The evaluator will attempt to modify all shared libraries that are used throughout the system.
Test Steps	Note: The TOE uses these permissions to protect the following from unauthorized modification: • Shared libraries – files in: o /usr/lib64/ o /usr/lib/

	1. Log into the TOE as an unprivileged user (user_unp). 2. Navigate to the directory /usr/lib 3. Attempt to modify files as an unprivileged user. 4. Verify the attempt failed. 5. Verify via logs that the attempts on modification have failed. 6. Log into the TOE as an unprivileged user (user_unp). 7. Navigate to the directory /usr/lib64 8. Attempt to modify files as an unprivileged user. 9. Verify the attempt failed. 10. Verify via logs that attempts on modification have failed.
Expected Test Results	The TOE should not allow an unprivileged user to modify the shared libraries.
Pass/Fail with Explanation	Pass. The TOE does not allow an unprivileged user to modify the shared libraries. This satisfies the testing requirement.

9.5.4 FPT_ACF_EXT.1 Test#36

Item	**Data**
Test Assurance Activity	The evaluator will create an unprivileged user account. Using this account, the evaluator will ensure that the following tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action): • **Test 36:** The evaluator will attempt to modify all system executables.
Test Steps	Note: The TOE uses these permissions to protect the following from unauthorized modification: • System executables – files in: ○ /usr/sbin/ ○ /usr/bin/ ○ /usr/libexec/ 1. Log into the TOE as an unprivileged user (user_unp). 2. Navigate to the directory /usr/bin 3. Attempt to modify files as an unprivileged user. 4. Verify the attempt failed. 5. Verify via logs that attempts on modification have failed. 6. Log into the TOE as an unprivileged user (user_unp). 7. Navigate to the directory /usr/libexec 8. Attempt to modify files as an unprivileged user. 9. Verify the attempt failed. 10. Verify via logs that attempts on modification have failed. 11. Log into the TOE as an unprivileged user (user_unp). 12. Navigate to the directory /usr/sbin 13. Attempt to modify files as an unprivileged user. 14. Verify the attempt failed. 15. Verify via logs that attempts on modification have failed.
Expected Test Results	The TOE should not allow an unprivileged user to modify the system executables.
Pass/Fail with Explanation	Pass. The TOE does not allow an unprivileged user to modify the system executables. This satisfies the testing requirement.

9.5.5 FPT_ACF_EXT.1 Test#37

Item	Data
Test Assurance Activity	The evaluator will create an unprivileged user account. Using this account, the evaluator will ensure that the following tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action): • **Test 37:** The evaluator will attempt to modify all system configuration files.
Test Steps	Note: The TOE uses these permissions to protect the following from unauthorized modification: • System configuration files – files in: ○ /etc/ ○ /usr/lib/ 1. Log into the TOE as an unprivileged user (user_unp). 2. Navigate to the directory /etc/ 3. Attempt to modify files as an unprivileged user. 4. Verify the attempt failed. 5. Verify via logs that attempts on modification have failed. 6. Log into the TOE as an unprivileged user (user_unp). 7. Navigate to the directory /usr/lib 8. Attempt to modify files as an unprivileged user. 9. Verify the attempt failed. 10. Verify via logs that attempts on modification have failed.
Expected Test Results	The TOE should not allow an unprivileged user to modify the system configuration files.
Pass/Fail with Explanation	Pass. The TOE does not allow an unprivileged user to modify the system configuration files. This satisfies the testing requirement.

9.5.6 FPT_ACF_EXT.1 Test#38

Item	Data
Test Assurance Activity	The evaluator will create an unprivileged user account. Using this account, the evaluator will ensure that the following tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action): • **Test 38:** The evaluator will attempt to modify any additional components selected.
Pass/Fail with Explanation	N/A. No additional components selected.

9.5.7 FPT_ACF_EXT.1 Test#39

Item	Data
Test Assurance Activity	The evaluator will create an unprivileged user account. Using this account, the evaluator will ensure that the following tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action):

	- **Test 39:** The evaluator will attempt to read security audit logs generated by the auditing subsystem
Test Steps	1. Log into the TOE as an unprivileged user (user_unp). 2. Navigate to the directory containing the security audit logs (/var/log/) and attempt to access them. 3. Verify that attempts to read security audit logs fail.
Expected Test Results	The TOE should not allow an unprivileged user to read the security audit logs.
Pass/Fail with Explanation	Pass. The TOE does not allow an unprivileged user to read the security audit logs. This satisfies the testing requirement.

9.5.8 FPT_ACF_EXT.1 Test#40

Item	Data
Test Assurance Activity	The evaluator will create an unprivileged user account. Using this account, the evaluator will ensure that the following tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action): - **Test 40:** The evaluator will attempt to read system-wide credential repositories
Test Steps	1. Log into the TOE as an unprivileged user. 2. Attempt to read system-wide credential repositories in the '/etc/shadow/' and '/etc/pki/ca-trust/source/anchors/' directory. 3. Verify that the attempt fails.
Expected Test Results	The TOE should not allow an unprivileged user to read the system-wide credential repositories.
Pass/Fail with Explanation	Pass. The TOE does not allow an unprivileged user to read system-wide credential repositories. This satisfies the testing requirements.

9.5.9 FPT_ACF_EXT.1 Test#41

Item	Data
Test Assurance Activity	The evaluator will create an unprivileged user account. Using this account, the evaluator will ensure that the following tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action): - **Test 41:** The evaluator will attempt to read any other object specified in the assignment
Pass/Fail with Explanation	N/A. No other object specified in the assignment.

9.5.10 FPT_ASLR_EXT.1

Item	Data
Test Assurance Activity	The evaluator will select 3 executables included with the TSF. If the TSF includes a web browser, it must be selected. If the TSF includes a mail client, it must be selected. For each of these apps, the evaluator will launch the same executables on two separate instances of the OS on identical hardware and compare all memory mapping locations. The evaluator will ensure that no memory mappings are placed in the same location. If the rare chance occurs that two mappings are the same for a single executable and not the same for

	the other two, the evaluator will repeat the test with that executable to verify that in the second test the mappings are different. This test can also be completed on the same hardware and rebooting between application launches.
Test Steps	Select 3 executables included with the TSF. (chronyd, auditd, fapolicyd) 1. Launch the executables and check the memory mapping for each one. 2. Reboot the system. 3. Launch the executables and verify that the mappings are different.
Expected Test Results	The TOE should not have executables running in the same memory locations.
Pass/Fail with Explanation	Pass. The TOE does not have executables running in the same memory locations. This satisfies the testing requirement.

9.5.11 FPT_SBOP_EXT.1 Test#42

Item	Data
Test Assurance Activity	For stack-based OSes, the evaluator will determine that the TSS contains a description of stackbased buffer overflow protections used by the OS. These are referred to by a variety of terms, such as stack cookie, stack guard, and stack canaries. The TSS must include a rationale for any binaries that are not protected in this manner. The evaluator will also preform the following test: • **Test 42:** The evaluator will inventory the kernel, libraries, and application binaries to determine those that do not implement stack-based buffer overflow protections. This list should match up with the list provided in the TSS. For OSes that store parameters/variables separately from control flow values, the evaluator will verify that the TSS describes what data structures control values, parameters, and variables are stored. The evaluator will also ensure that the TSS includes a description of the safeguards that ensure parameters and variables do not intermix with control flow values.
Test Steps	1. Run annocheck on files in '/usr/lib/firmware/' directory. 2. Verify the files do not implement stack-based buffer overflow protections. 3. Run annocheck on files in '/usr/lib64/gconv/' directory. 4. Verify the files do not implement stack-based buffer overflow protections. 5. Run annocheck on files in '/usr/lib64/' directory. 6. Verify files mentioned in the TSS fail the protection check. 7. Run annocheck on files in '/usr/sbin/' directory. 8. Verify files mentioned in the TSS fail the protection check. 9. Run annocheck on files in '/usr/lib/gcc/x86_64-redhat-linux/8' directory. 10. Verify the files mentioned in TSS fail the protection check. 11. Run annocheck on '/usr/lib/grub/i386-pc/kernel.exec' 12. Verify the files do not implement stack-based buffer overflow protections.
Expected Test Results	The TOE should be implementing stack-based buffer overflow protections for kernel, libraries, and application binaries except the ones mentioned in the TSS

Pass/Fail with Explanation	Pass. The TOE implements stack-based buffer overflow protections for kernel, libraries, and application binaries except the ones mentioned in the TSS.

9.5.12 FPT_SRP_EXT.1 Test#84

Item	**Data**
Test Assurance Activity	Test 84[conditional, to be performed if file path is selected from FPT_SRP_EXT.1.1]: • The evaluator will configure the OS to only allow code execution from the core OS directories. The evaluator will then attempt to execute code from a directory that is in the allowed list. The evaluator will ensure that the code they attempted to execute has been executed.
Test Steps	1. The evaluator will ensure that the TOE is configured to allow code execution from the core OS directories. 2. Attempt to execute code from a directory that is in the allowed list. 3. Verify that the execution succeeded.
Expected Test Results	The TOE should allow code to execute in directories where it is allowed.
Pass/Fail with Explanation	Pass. The TOE allows code to be executed in directories where it is allowed. This satisfies the testing requirement.

9.5.13 FPT_SRP_EXT.1 Test#85

Item	**Data**
Test Assurance Activity	Test 85[conditional, to be performed if file path is selected from FPT_SRP_EXT.1.1]: • The evaluator will configure the OS to only allow code execution from the core OS directories. The evaluator will then attempt to execute code from a directory that is not in the allowed list. The evaluator will ensure that the code they attempted to execute has not been executed
Test Steps	1. The evaluator will ensure that the TOE is configured to allow code execution from the core OS directories. 2. Attempt to execute code from a directory that is not in the allowed list. 3. Verify that the execution failed.
Expected Test Results	The TOE should allow code to execute in directories where it is allowed.
Pass/Fail with Explanation	Pass. The TOE does not allow code to be executed from a directory that is not in the allowed list. This satisfies the testing requirement.

9.5.14 FPT_SRP_EXT.1 Test#86

Item	**Data**
Test Assurance Activity	Test 86[conditional, to be performed if file digital signature is selected from FPT_SRP_EXT.1.1]: • The evaluator will configure the OS to only allow code that has been signed by the OS vendor to execute. The evaluator will then attempt

	to execute code signed by the OS vendor. The evaluator will ensure that the code they attempted to execute has been executed.
Pass/Fail with Explanation	N/A, because the option "file digital signature" is not selected in the SFR: FPT_SRP_EXT.1.1.

9.5.15 FPT_SRP_EXT.1 Test#87

Item	**Data**
Test Assurance Activity	Test 87[conditional, to be performed if file digital signature is selected from FPT_SRP_EXT.1.1]: • The evaluator will configure the OS to only allow code that has been signed by the OS vendor to execute. The evaluator will then attempt to execute code signed by another digital authority. The evaluator will ensure that the code they attempted to execute has not been executed.
Pass/Fail with Explanation	N/A, because the option "file digital signature" is not selected in the SFR: FPT_SRP_EXT.1.1.

9.5.16 FPT_SRP_EXT.1 Test#88

Item	**Data**
Test Assurance Activity	Test 88[conditional, to be performed if version is selected from FPT_SRP_EXT.1.1]: • The evaluator will configure the OS to allow execution of a specific application based on version. The evaluator will then attempt to execute the same version of the application. The evaluator will ensure that the code they attempted to execute has been executed.
Pass/Fail with Explanation	N/A, because the option "version" is not selected in the SFR: FPT_SRP_EXT.1.1.

9.5.17 FPT_SRP_EXT.1 Test#89

Item	**Data**
Test Assurance Activity	Test 89[conditional, to be performed if version is selected from FPT_SRP_EXT.1.1]: • The evaluator will configure the OS to allow execution of a specific application based on version. The evaluator will then attempt to execute an older version of the application. The evaluator will ensure that the code they attempted to execute has not been executed.
Pass/Fail with Explanation	N/A, because the option "version" is not selected in the SFR: FPT_SRP_EXT.1.1.

9.5.18 FPT_SRP_EXT.1 Test#90

Item	**Data**
Test Assurance Activity	Test 90[conditional, to be performed if hash is selected from FPT_SRP_EXT.1.1]:

	- The evaluator will configure the OS to allow execution based on the hash of the application executable. The evaluator will then attempt to execute the application with the matching hash. The evaluator will ensure that the code they attempted to execute has been executed.
Pass/Fail with Explanation	N/A, because the option “Hash” is not selected in the SFR: FPT_SRP_EXT.1.1.

9.5.19 FPT_SRP_EXT.1 Test#91

Item	**Data**
Test Assurance Activity	Test 91[conditional, to be performed if hash is selected from FPT_SRP_EXT.1.1]: - The evaluator will configure the OS to allow execution based on the hash of the application executable. The evaluator will modify the application in such a way that the application hash is changed. The evaluator will then attempt to execute the application with the matching hash. The evaluator will ensure that the code they attempted to execute has not been executed.
Pass/Fail with Explanation	N/A, because the option “Hash” is not selected in the SFR: FPT_SRP_EXT.1.1.

9.5.20 FPT_SRP_EXT.1 Test#92

Item	**Data**
Test Assurance Activity	Test 92[conditional, to be performed if other is selected from FPT_SRP_EXT.1.1]: - The evaluator will attempt to run an application that should be allowed based on the defined software restriction policy and ensure that it runs.
Pass/Fail with Explanation	N/A, because the option “other” is not selected in the SFR: FPT_SRP_EXT.1.1.

9.5.21 FPT_SRP_EXT.1 Test#93

Item	**Data**
Test Assurance Activity	Test 93[conditional, to be performed if other is selected from FPT_SRP_EXT.1.1]: - The evaluator will then attempt to run an application that should not be allowed the defined software restriction policy and ensure that it does not run.
Pass/Fail with Explanation	N/A, because the option “other” is not selected in the SFR: FPT_SRP_EXT.1.1.

9.5.22 FPT_TST_EXT.1 Test#43

Item	**Data**
Test Assurance Activity	The evaluator will also perform the following test:

	- Test 43: The evaluator will perform actions to cause TSF software to load and observe that the integrity mechanism does not flag any executables as containing integrity errors and that the OS properly boots.
Test Steps	1. Reboot the TOE. 2. Verify there are no errors in the boot process. 3. Verify that there are no errors in the log
Expected Test Results	The TOE should not flag any executables as containing integrity errors and boot properly.
Pass/Fail with Explanation	Pass. The TOE did not show any integrity errors and boot process was performed correctly. This satisfies the testing requirement.

9.5.23 FPT_TST_EXT.1 Test#44

Item	Data
Test Assurance Activity	The evaluator will also preform the following test: - Test 44: The evaluator will modify a TSF executable that is part of the bootchain verified by the TSF (i.e. Not the first-stage bootloader) and attempt to boot. The evaluator will ensure that an integrity violation is triggered and the OS does not boot (Care must be taken so that the integrity violation is determined to be the cause of the failure to load the module, and not the fact that in such a way to invalidate the structure of the module.).
Test Steps	1. Modify a boot file 2. Reboot the TOE 3. Verify that the boot fails
Expected Test Results	The TOE should not boot if a boot file is modified.
Pass/Fail with Explanation	Pass. The TOE does not boot if a boot file is modified. This satisfies the testing requirement.

9.5.24 FPT_TST_EXT.1 Test#45

Item	Data
Test Assurance Activity	The evaluator will also perform the following test: - Test 45[conditional, to be performed if a digital signature using an X509 certificate with hardware-based protection is selected from FPT_TST_EXT.1.1]: If the ST author indicates that the integrity verification is performed using public key in an X509 certificate, the evaluator will verify that the boot integrity mechanism includes a certificate validation according to FIA_X509_EXT.1 for all certificates in the chain from the certificate used for boot integrity to a certificate in the trust store that are not themselves in the trust store. This means that, for each X509 certificate in this chain that is not a trust store element, the evaluator must ensure that revocation information is available to the TOE during the bootstrap mechanism (before the TOE becomes fully operational).
Pass/Fail with Explanation	N/A, because the option :" digital signature using an X509 certificate with hardware-based protection" is not selected from FPT_TST_EXT.1.1.

9.5.25 FPT_TUD_EXT.1 Test#46

Item	Data
Test Assurance Activity	The evaluator will check for an update using procedures described in the documentation and verify that the OS provides a list of available updates. Testing this capability may require installing and temporarily placing the system into a configuration in conflict with secure configuration guidance which specifies automatic update. The evaluator is also to ensure that the response to this query is authentic by using a digital signature scheme specified in FCS_COP.1/SIGN. The digital signature verification may be performed as part of a network protocol occurs over a trusted channel as described in FTP_ITC_EXT.1.) If the signature verification is not performed as part of a trusted channel, the evaluator will send a query response with a bad signature and verify that the signature verification fails. The evaluator will then send a query response with a good signature and verify that the signature verification is successful. For the following tests, the evaluator will initiate the download of an update and capture the update prior to installation. The download could originate from the vendor's website, an enterprise-hosted update repository, or another system (e.g. network peer). All supported origins for the update must be indicated in the TSS and evaluated. • Test 46: The evaluator will ensure that the update has a digital signature belonging to the vendor prior to its installation. The evaluator will modify the downloaded update in such a way that the digital signature is no longer valid. The evaluator will then attempt to install the modified update. The evaluator will ensure that the OS does not install the modified update.
Test Steps	1. Check for the OS updates from a source that has a good digital signature and verify that the check succeeds. 2. Verify the key matches with the one imported on the TOE. 3. Import a new key to fail the upgrade check step. 4. Check for update from the source after the key has been replaced. 5. Verify that the check fails.
Expected Test Results	The TOE should properly check updates from a source that has a good signature. The TOE should not be able to check for updates from a source that has a bad signature.
Pass/Fail with Explanation	Pass. The TOE properly checks updates from a source that has a good digital signature. This satisfies the testing requirement.

9.5.26 FPT_TUD_EXT.1 Test#47

Item	Data
Test Assurance Activity	The evaluator will check for an update using procedures described in the documentation and verify that the OS provides a list of available updates. Testing this capability may require installing and temporarily placing the system into a configuration in conflict with secure configuration guidance which specifies automatic update.

	The evaluator is also to ensure that the response to this query is authentic by using a digital signature scheme specified in FCS_COP.1/SIGN. The digital signature verification may be performed as part of a network protocol occurs over a trusted channel as described in FTP_ITC_EXT.1.) If the signature verification is not performed as part of a trusted channel, the evaluator will send a query response with a bad signature and verify that the signature verification fails. The evaluator will then send a query response with a good signature and verify that the signature verification is successful. For the following tests, the evaluator will initiate the download of an update and capture the update prior to installation. The download could originate from the vendor's website, an enterprise-hosted update repository, or another system (e.g. network peer). All supported origins for the update must be indicated in the TSS and evaluated. • Test 47: The evaluator will ensure that the update has a digital signature belonging to the vendor. The evaluator will then attempt to install the update (or permit installation to continue). The evaluator will ensure that the OS successfully installs the update.
Test Steps	1. Check for authentic TOE's update package. 2. Initiate update using the authentic TOE's update package. 3. Verify that the OS successfully installed the update.
Expected Test Results	The TOE should be able to install an authentic OS update.
Pass/Fail with Explanation	Pass. The TOE was able to install an authentic OS update. This satisfies the testing requirement.

9.5.27 FPT_TUD_EXT.2 Test#48

Item	Data
Test Assurance Activity	The evaluator will check for updates to application software using procedures described in the documentation and verify that the OS provides a list of available updates. Testing this capability may require temporarily placing the system into a configuration in conflict with secure configuration guidance which specifies automatic update. The evaluator is also to ensure that the response to this query is authentic by using a digital signature scheme specified in FCS_COP.1/SIGN. The digital signature verification may be performed as part of a network protocol as described in FTP_ITC_EXT.1. If the signature verification is not performed as part of a trusted channel, the evaluator will send a query response with a bad signature and verify that the signature verification fails. The evaluator will then send a query response with a good signature and verify that the signature verification is successful. The evaluator will initiate an update to an application. This may vary depending on the application, but it could be through the application vendor's website, a commercial app store, or another system. All origins supported by the OS must be indicated in the TSS and evaluated. However, this only includes those mechanisms for which the OS is providing a trusted installation and update functionality. It does not include user or administrator-driven download and installation of arbitrary files.

	• Test 48: The evaluator will ensure that the update has a digital signature which chains to the OS vendor or another trusted root managed through the OS. The evaluator will modify the downloaded update in such a way that the digital signature is no longer valid. The evaluator will then attempt to install the modified update. The evaluator will ensure that the OS does not install the modified update
Test Steps	1. Check for the application updates from a source that has a good digital signature and verify that the check succeeds. 2. Verify the key matches with the one imported on the TOE. 3. Import a new key to fail the upgrade check step. 4. Check for update from the source after the key has been replaced. 5. Verify that the check fails.
Expected Test Results	The TOE should properly check updates for application software from a source that has a good signature. The TOE should not be able to check for updates for applications from a source that has a bad signature.
Pass/Fail with Explanation	Pass. The TOE properly checks updates from a source that has a good digital signature. This satisfies the testing requirement.

9.5.28 FPT_TUD_EXT.2 Test#49

Item	**Data**
Test Assurance Activity	The evaluator will check for updates to application software using procedures described in the documentation and verify that the OS provides a list of available updates. Testing this capability may require temporarily placing the system into a configuration in conflict with secure configuration guidance which specifies automatic update. The evaluator is also to ensure that the response to this query is authentic by using a digital signature scheme specified in FCS_COP.1/SIGN. The digital signature verification may be performed as part of a network protocol as described in FTP_ITC_EXT.1. If the signature verification is not performed as part of a trusted channel, the evaluator will send a query response with a bad signature and verify that the signature verification fails. The evaluator will then send a query response with a good signature and verify that the signature verification is successful. The evaluator will initiate an update to an application. This may vary depending on the application, but it could be through the application vendor's website, a commercial app store, or another system. All origins supported by the OS must be indicated in the TSS and evaluated. However, this only includes those mechanisms for which the OS is providing a trusted installation and update functionality. It does not include user or administrator-driven download and installation of arbitrary files. Test 49: The evaluator will ensure that the update has a digital signature belonging to the OS vendor or another trusted root managed through the OS. The evaluator will then attempt to install

	the update. The evaluator will ensure that the OS successfully installs the update.
Test Steps	1. Check for authentic application update package. 2. Initiate update of authentic application update package. 3. Verify that the OS successfully installed the application update.
Expected Test Results	The TOE should be able to install an authentic application software update.
Pass/Fail with Explanation	Pass. The TOE was able to install an authentic application software update .This satisfies the testing requirement.

9.6 FIA

9.6.1 FIA_AFL.1 Test#53 (TD0691)

Item	**Data**
Test Assurance Activity	The evaluator will set an administrator-configurable threshold for failed attempts, or note the ST-specified assignment. The evaluator will then (per selection) repeatedly attempt to authenticate with an incorrect password, PIN, or certificate until the number of attempts reaches the threshold. Note that the authentication attempts and lockouts must also be logged as specified in FAU_GEN.1. Test 53 [**conditional, to be performed if "authentication based on user name and password" is selected in FIA_AFL.1 and FIA_UAU.5**]: The evaluator will attempt to authenticate repeatedly to the system with a known bad password. Once the defined number of failed authentication attempts has been reached the evaluator will ensure that the account that was being used for testing has had the actions detailed in the assignment list above applied to it. The evaluator will ensure that an event has been logged to the security event log detailing that the account has had these actions applied.
Test Steps	1. Set user login unsuccessful authentication attempts to 3 attempts. 2. Verify configuration of unsuccessful authentication attempt via logs. 3. Start a local session with the TOE and attempt to login 3 times with wrong password and lockout the user. 4. Verify the user is locked out for configured time via logs. 5. Attempt to open another connection and attempt to login with valid password before the lockout period expires. 6. Verify with logs the attempt failed due to lockout account.
Expected Test Results	The TOE should properly react (as selected in the ST) to a user that has reached the configured limit for consecutive failed authentication attempts and create an audit log.
Pass/Fail with Explanation	Pass. The TOE reacts as specified in the ST when the evaluator reaches the configured limit for consecutive failed authentication attempts and create an audit log. This satisfies the testing requirement.

9.6.2 FIA_AFL.1 Test#54 (TD0691)

Item	**Data**

Test Assurance Activity	The evaluator will set an administrator-configurable threshold for failed attempts, or note the ST-specified assignment. The evaluator will then (per selection) repeatedly attempt to authenticate with an incorrect password, PIN, or certificate until the number of attempts reaches the threshold. Note that the authentication attempts and lockouts must also be logged as specified in FAU_GEN.1. Test 54 [**conditional, to be performed if "authentication based on user name and a PIN that releases an asymmetric key stored in OE-protected storage" is selected in FIA_AFL.1 and FIA_UAU.5**]: The evaluator will attempt to authenticate repeatedly to the system with a known bad PIN. Once the defined number of failed authentication attempts has been reached the evaluator will ensure that the account that was being used for testing has had the actions detailed in the assignment list above applied to it. The evaluator will ensure that an event has been logged to the security event log detailing that the account has had these actions applied.
Pass/Fail with Explanation	N/A, because the option "authentication based on user name and a PIN that releases an asymmetric key stored in OE-protected storage" is not selected in FIA_AFL.1 nor in FIA_UAU.5.

9.6.3 FIA_AFL.1 Test#55 (TD0691)

Item	Data
Test Assurance Activity	The evaluator will set an administrator-configurable threshold for failed attempts, or note the ST-specified assignment. The evaluator will then (per selection) repeatedly attempt to authenticate with an incorrect password, PIN, or certificate until the number of attempts reaches the threshold. Note that the authentication attempts and lockouts must also be logged as specified in FAU_GEN.1. Test 55 [**conditional, to be performed if "authentication based on X.509 certificates" is selected in FIA_AFL.1 and FIA_UAU.5**]: The evaluator will attempt to authenticate repeatedly to the system using a known bad certificate. Once the defined number of failed authentication attempts has been reached the evaluator will ensure that the account that was being used for testing has had the actions detailed in the assignment list above applied to it. The evaluator will ensure that an event has been logged to the security event log detailing that the account has had these actions applied.
Pass/Fail with Explanation	N/A, because the option:' authentication based on X.509 certificates" is not selected in FIA_AFL.1 nor FIA_UAU.5.".

9.6.4 FIA_UAU.5.1 Test#56

Item	Data
Test Assurance Activity	The following content should be included if: • authentication based on username and password is selected from FIA_UAU.5.1 • Test 56: The evaluator will attempt to authenticate to the OS using the known username and password. The evaluator will ensure that the authentication attempt is successful.
Test Steps	1. Attempt to login with correct username/password.

	2. Verify via logs that authentication attempt is successful.
Expected Test Results	The TOE should allow the evaluator to login when correct user credentials are used.
Pass/Fail with Explanation	Pass. The TOE allows the evaluator to login when correct user credentials are used. This satisfies the testing requirement.

9.6.5 FIA_UAU.5.1 Test#57

Item	Data
Test Assurance Activity	The following content should be included if: • authentication based on username and password is selected from FIA_UAU.5.1 • Test 57: The evaluator will attempt to authenticate to the OS using the known user name but an incorrect password. The evaluator will ensure that the authentication attempt is unsuccessful.
Test Steps	1. Attempt to login with correct username and incorrect password. 2. Verify via logs that authentication attempt is unsuccessful.
Expected Test Results	The TOE should not allow the evaluator to login when incorrect user credentials are used.
Pass/Fail with Explanation	Pass. The TOE does not allow the evaluator to login when incorrect user credentials are used. This satisfies the testing requirement.

9.6.6 FIA_UAU.5.1 Test#58

Item	Data
Test Assurance Activity	The following content should be included if: • username and a PIN that releases an asymmetric key is selected from FIA_UAU.5.1 The evaluator will examine the TSS for guidance on supported protected storage and will then configure the TOE or OE to establish a PIN which enables release of the asymmetric key from the protected storage (such as a TPM, a hardware token, or isolated execution environment) with which the OS can interface. The evaluator will then conduct the following tests: • Test 58: The evaluator will attempt to authenticate to the OS using the known user name and PIN. The evaluator will ensure that the authentication attempt is successful
Pass/Fail with Explanation	N/A. because the option "authentication based on username and a PIN that releases an asymmetric key" is not selected in FIA_UAU.5

9.6.7 FIA_UAU.5.1 Test#59

Item	Data
Test Assurance Activity	The following content should be included if: • username and a PIN that releases an asymmetric key is selected from FIA_UAU.5.1 The evaluator will examine the TSS for guidance on supported protected storage and will then configure the TOE or OE to establish a PIN which enables

	release of the asymmetric key from the protected storage (such as a TPM, a hardware token, or isolated execution environment) with which the OS can interface. The evaluator will then conduct the following tests: - Test 59: The evaluator will attempt to authenticate to the OS using the known user name but an incorrect PIN. The evaluator will ensure that the authentication attempt is unsuccessful
Pass/Fail with Explanation	N/A, because the option "authentication based on username and a PIN that releases an asymmetric key" is not selected in FIA_UAU.5

9.6.8 FIA_UAU.5.1 Test#60

Item	Data
Test Assurance Activity	The following content should be included if: - combination of authentication based on user name, password, and time-based one-time password is selected from FIA_UAU.5.1 The evaluator will configure the OS to authentication to authenticate to the OS using a username, password, and one-time password mechanism. The evaluator will then perform the following tests. - Test 60: The evaluator will attempt to authenticate using a valid username, valid password, and valid one-time password. The evaluator will ensure that the authentication attempt is successful.
Pass/Fail with Explanation	N/A, because the option “combination of authentication based on username, password, and time-based one-time password.” is not selected from FIA_UAU.5.1.

9.6.9 FIA_UAU.5.1 Test#61

Item	Data
Test Assurance Activity	The following content should be included if: - combination of authentication based on user name, password, and time-based one-time password is selected from FIA_UAU.5.1 The evaluator will configure the OS to authentication to authenticate to the OS using a username, password, and one-time password mechanism. The evaluator will then perform the following tests. - Test 61: The evaluator will attempt to authenticate using a valid username, invalid password, and valid one-time password. The evaluator will ensure that the authentication attempt fails.
Pass/Fail with Explanation	N/A, because the option “combination of authentication based on username, password, and time-based one-time password.” is not selected from FIA_UAU.5.1.

9.6.10 FIA_UAU.5.1 Test#62

Item	Data
Test Assurance Activity	The following content should be included if: - combination of authentication based on user name, password, and time-based one-time password is selected from FIA_UAU.5.1

	The evaluator will configure the OS to authentication to authenticate to the OS using a username, password, and one-time password mechanism. The evaluator will then perform the following tests. • Test 62: The evaluator will attempt to authenticate using a valid username, valid password, and invalid one-time password. The evaluator will ensure that the authentication attempt fails.
Pass/Fail with Explanation	N/A, because the option "combination of authentication based on username, password, and time-based one-time password." is not selected from FIA_UAU.5.1.

9.6.11 FIA_UAU.5.1 Test#63

Item	Data
Test Assurance Activity	The following content should be included if: • combination of authentication based on user name, password, and time-based one-time password is selected from FIA_UAU.5.1 The evaluator will configure the OS to authentication to authenticate to the OS using a username, password, and one-time password mechanism. The evaluator will then perform the following tests. • Test 63: The evaluator will attempt to authenticate using a valid username, invalid password, and invalid one-time password. The evaluator will ensure that the authentication attempt fails.
Pass/Fail with Explanation	N/A, because the option "combination of authentication based on username, password, and time-based one-time password." is not selected from FIA_UAU.5.1.

9.6.12 FIA_X509_EXT.1 Test#64

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • Test 64: The evaluator will 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 will then establish a valid certificate path consisting of valid CA certificates, and demonstrate that the function succeeds. The evaluator will then remove trust in one of the CA certificates, and show that the function fails.
Test Steps	Establishing a certificate path in which one of the issuing certificates is not a CA certificate, by setting the basicConstraints field in an issuing certificate to have CA flag=False. 1. Generate a chain of 4 certificates with one of the certificates having CA flag=False. 2. Attempt a connection from the TOE with the OpenSSL server and verify the connection fails. 3. Verify the connection with Packet capture. Omitting the basicConstraints field in one of the issuing certificates. 1. Generate a chain of 4 certificates with one of the certificates missing basicConstraints field. 2. Attempt a connection from the TOE with the OpenSSL server and verify the connection fails. 3. Verify the connection with Packet capture. Omitting the CA signing bit of the key usage field in an issuing certificate. 1. Generate a chain of 4 certificates with certificate ICA2 missing CA signing bit of the key usage field. 2. Attempt a connection from the TOE with the OpenSSL server and verify the connection fails. 3. Verify the connection with Packet capture. Setting the path length field of a valid CA field to a value strictly less than the certificate path. 1. Generate a chain of 4 certificates with certificates having CA field to a value strictly less than the certificate path. 2. Attempt a connection from the TOE with the OpenSSL server and verify the connection fails. 3. Verify the connection with Packet capture. Valid certificate path 1. The evaluator generated a chain of 4 certificates. (Root_CA->ICA1->ICA2->Endcert) 2. Delete ICA2 certificate from the TOE System keychain. 3. Attempt a connection from the TOE with the OpenSSL server and verify the connection fails. 4. Load the missing ICA2 certificate into the TOE keychain. 5. Attempt a connection from the TOE with the OpenSSL server and verify the connection is successful.
Expected Test Results	The TOE should not connect to a TLS server when using a certificate path in which one of the issuing certificates is not a CA certificate.

	The TOE should not connect to a TLS server when setting the basicConstraints field in an issuing certificate to have CA=False. The TOE should not connect to a TLS server when omitting the basicConstraints field in one of the issuing certificates. The TOE should not connect to a TLS server when omitting the CA signing bit of the key usage field in an issuing certificate. The TOE should not connect to a TLS server when setting the path length field of a valid CA field to a value strictly less than the certificate path. The TOE should connect to a TLS server when a valid certificate path consisting of valid CA certificates is used. The TOE should not connect to a TLS server when trust in one of the CA certificates is removed.
Pass/Fail with Explanation	Pass. The evaluator observed that the TOE rejects a certificate without a valid certification path resulting in the communications channel not being established with the TLS server, for each of the following reasons: • 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. When all proper X509 conditions were met the evaluator observed the connection was successfully established. This meets the testing requirements.

9.6.13 FIA_X509_EXT.1 Test#65

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • Test 65**:** The evaluator will demonstrate that validating an expired certificate results in the function failing.
Test Steps	1. Start a TLS session with a certificate that has expired. 2. Attempt to connect from the TOE to the TLS server. 3. Verify that the connection fails via packet capture.
Expected Test Results	The TOE should not connect to a TLS server if the server certificate has expired.
Pass/Fail with Explanation	Pass. The TOE does not connect to a TLS server if the server certificate has expired. This satisfies the testing requirement.

9.6.14 FIA_X509_EXT.1 Test#66 (TD0773)

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • Test 66: [Conditional, to be performed for use cases identified in exceptions that cannot be configured to allow revocation checking] The evaluator will test that the OS can properly handle revoked certificates - conditional on whether CRL, OCSP, OCSP stapling, or OCSP multi-stapling is selected; if multiple methods are selected, then a test will be performed for each method. The evaluator will test revocation of the node certificate and revocation of the intermediate CA certificate (i.e. the intermediate CA certificate should be revoked by the root CA). If OCSP stapling per RFC 6066 is the only supported revocation method, testing revocation of the intermediate CA certificate is omitted. The evaluator will 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 If the exceptions are configurable, the evaluator shall attempt to configure the exceptions to allow revocation checking for each function indicated in FIA_X509_EXT.2.
Test Steps	Valid ICA and valid leaf certificate: 1. Start a TLS session with a valid ICA. 2. Attempt to connect via TLS from the TOE. 3. Verify that the connection succeeds via packet capture. Revoked Leaf Certificate: 1. Start a TLS session with a revoked leaf certificate. 2. Attempt to connect via TLS from the TOE. 3. Verify that the connection fails via packet capture. Revoked ICA, Valid Leaf Certificate: 1. Start a TLS session with a revoked ICA and a valid leaf certificate. 2. Attempt to connect via TLS from the TOE. 3. Verify that the connection fails via packet capture.
Expected Test Results	The TOE should not connect to the TLS server when either the node certificate or intermediate certificate is revoked.
Pass/Fail with Explanation	Pass. The TOE does not include exceptions and validates the revocation status of the certificate using CRL. The TOE properly handles a revoked server or intermediate CA cert and connects when the certs are not revoked. This satisfies the testing requirements.

9.6.15 FIA_X509_EXT.1 Test#67

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. - Test 67: If any OCSP option is selected, the evaluator will configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator will configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.
Test Steps	Note: The OCSP option is not selected in ST. 1. Start a TLS server with a certificate signed by the ICA that does not have a cRLsign key usage bit. 2. Attempt to connect to the TLS server from the TOE. 3. Verify that the connection fails using packet capture.
Expected Test Results	The TOE should not connect to the TLS server if the CA that signs the CRL does not have the cRLsign key.
Pass/Fail with Explanation	Pass. The OCSP option is not selected in the ST. The validation of the CRL failed with a CA certificate lacking the CRLsign bit, and the TLS connection was rejected. This meets the testing requirements.

9.6.16 FIA_X509_EXT.1 Test#68

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. - Test 68: The evaluator will 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	1. Start the acumen-tlsc tool to modify any byte in the first eight bytes of the certificate. 2. Attempt to connect to the acumen-tlsc tool from the TOE with the modified certificate. 3. Verify that the connection fails via packet capture.
Expected Test Results	The TOE should not connect to a TLS server with a server certificate that had the first eight bytes modified.
Pass/Fail with Explanation	Pass. The evaluator modified the first eight bytes of the certificate being presented by the server and ensured that the certificate fails to validate, and the TLS handshake fails. This satisfies the testing requirement.

9.6.17 FIA_X509_EXT.1 Test#69 (TD0773)

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • Test 69: The evaluator will modify any byte in the last eight bytes of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)
Test Steps	1. Start the acumen-tlsc tool to modify any byte in the last eight bytes of the certificate. 2. Attempt to connect to the acumen-tlsc tool from the TOE. 3. Verify that the connection fails via packet capture.
Expected Test Results	The TOE should not connect to a TLS server with a server certificate that had the last eight bytes modified.
Pass/Fail with Explanation	Pass. The evaluator modified the last byte of the certificate and demonstrated that the certificate fails to validate. This satisfies the testing requirement.

9.6.18 FIA_X509_EXT.1 Test#70

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • Test 70: The evaluator will modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The signature of the certificate will not validate.)
Test Steps	1. Start the acumen-tlsc tool to modify any byte in the public key of the certificate. 2. Attempt to connect to the acumen-tlsc tool from the TOE. 3. Verify that the connection fails via packet capture.
Expected Test Results	The TOE should not connect to a TLS server with a server certificate that had the public key modified.
Pass/Fail with Explanation	Pass. The evaluator modified 8 bytes in the public key of the server certificate and demonstrated that the certificate fails to validate. This satisfies the testing requirement.

9.6.19 FIA_X509_EXT.1 Test#71.1

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • Test 71[conditional, to be performed if ○ ECDSA schemes is selected from FCS_COP.1.1/SIGN ○ 6187 is selected from FCS_SSH_EXT.1.1 from Functional Package for Secure Shell (SSH), version 1.0

	○ Test 71.1: The evaluator will 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 will confirm that the TOE validates the certificate chain.
Test Steps	1. Show EC certificates used in for TLS session (root, ICA1, ICA2, and leaf). 2. Start a TLS session that is using all EC certificates. 3. Attempt to connect to the TLS server from the TOE using an EC certificate. 4. Verify that the connection succeeds via packet capture.
Expected Test Results	The TOE should connect to a TLS server when using an EC certificate.
Pass/Fail with Explanation	Pass. The TOE connects to a TLS server when using an EC certificate. This satisfies the testing requirement.

9.6.20 FIA_X509_EXT.1 Test#71.2

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • Test 71[conditional, to be performed if ○ ECDSA schemes is selected from FCS_COP.1.1/SIGN ○ 6187 is selected from FCS_SSH_EXT.1.1 from Functional Package for Secure Shell (SSH), version 1.0 ○ Test 71.2: The evaluator will replace the intermediate certificate in the certificate chain for Test 71.1 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 71.1, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator will confirm the TOE treats the certificate as invalid.
Test Steps	1. Modify the ICA certificate from FIA_X509_EXT.1, Test#71.1 using x509-mod tool. 2. Start a TLS session using the new explicit certificate. 3. Attempt to connect to the TLS server from the TOE. 4. Verify that the connection fails via packet capture.
Expected Test Results	The TOE should not connect to a TLS server with an intermediate EC certificate modified to have the public key information field where the EC parameters use an explicit format version of the Elliptic Curve parameters in the public key information field. This satisfies the testing requirement.

Pass/Fail with Explanation	Pass. The TOE does not connect to a TLS server with an intermediate EC certificate modified to have the public key information field where the EC parameters use an explicit format version of the Elliptic Curve parameters in the public key information field. This satisfies the testing requirement.

9.6.21 FIA_X509_EXT.1 Test#72

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • Test 72[conditional, to be performed if exceptional use cases is selected from FIA_X509_EXT.1.1]: For each exceptional use case for revocation checking described in the ST, the evaluator shall attempt to establish the conditions of the use case, designate the certificate as invalid and perform the function relying on the certificate. The evaluator shall observe that the alternate revocation checking mechanism successfully prevents performance of the function.
Pass/Fail with Explanation	N/A, because there are no exceptional use cases selected in the SFR:FIA_X509_EXT.1.1 in the ST.

9.6.22 FIA_X509_EXT.1 Test#73 (TD0773)

Item	**Data**
Test Assurance Activity	[Conditional, to be performed if "authentication based on X.509 certificates" is selected in FIA_UAU.5]: The evaluator will generate an X.509v3 certificate for a user with the Client Authentication Extended Key Usage field set. The evaluator will provision the OS for authentication with the X.509v3 certificate. The evaluator will ensure that the certificates are validated by the OS as per FIA_X509_EXT.1.1 and then conduct the following two tests: • Test 73: The evaluator will attempt to authenticate to the OS using the X.509v3 certificate. The evaluator will ensure that the authentication attempt is successful.
Pass/Fail with Explanation	N/A, because the option "authentication based on X.509 certificates" is not selected in FIA_UAU.5 in ST.

9.6.23 FIA_X509_EXT.1 Test#74 (TD0773)

Item	**Data**
Test Assurance Activity	[Conditional, to be performed if "authentication based on X.509 certificates" is selected in FIA_UAU.5]: The evaluator will generate an X.509v3 certificate for a user with the Client Authentication Extended Key Usage field set. The evaluator will provision the OS for authentication with the X.509v3

	certificate. The evaluator will ensure that the certificates are validated by the OS as per FIA_X509_EXT.1.1 and then conduct the following two tests: • Test 74: The evaluator will generate a second certificate identical to the first except for the public key and any values derived from the public key. The evaluator will attempt to authenticate to the OS with this certificate. The evaluator will ensure that the authentication attempt is unsuccessful.
Pass/Fail with Explanation	N/A, because the option "authentication based on X.509 certificates" is not selected in FIA_UAU.5 in ST.

9.6.24 FIA_X509_EXT.1 Test#75

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • Test 75: The evaluator will construct a certificate path, such that the certificate of the CA issuing the OS's certificate does not contain the basicConstraints extension. The validation of the certificate path fails.
Pass/Fail with Explanation	Pass. Covered in FIA_X509_EXT.1.1 Test 64, as the TOE will not validate a certificate with missing basicConstraints inside an issuer's certificate, but it will accept that same certificate when it has the full CA chain with the basicConstraints field defined in the issuing certificates. Incomplete certificates (without the basicConstraints extension) fail to validate and are rejected. This satisfies the testing requirements.

9.6.25 FIA_X509_EXT.1 Test#76

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • Test 76: The evaluator will construct a certificate path, such that the certificate of the CA issuing the OS's certificate has the CA flag in the basicConstraints extension not set. The validation of the certificate path fails.
Pass/Fail with Explanation	Pass. Covered in FIA_X509_EXT.1 Test#64 as the TOE will not validate a certificate with missing CA flag inside an issuer's certificate, but it will accept that same certificate when it has the full CA chain with the CA flag set inside the issuing certificates. This satisfies the testing requirements.

9.6.26 FIA_X509_EXT.1 Test#77

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 evaluator will create a chain of at least four certificates: the node certificate to be tested, two Intermediate CAs, and the self-signed Root CA. • Test 77: The evaluator will construct a certificate path, such that the certificate of the CA issuing the OS's certificate has the CA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.
Pass/Fail with Explanation	Pass. Covered in FCS_TLSC_EXT.1.1.Test#1. The validation of the certificate path succeeds when CA issuing certificates (Root_CA, ICA1 and ICA2) have basicConstraints extension set to TRUE. This satisfies the testing requirement.

9.6.27 FIA_X509_EXT.2 (TD0789)

Item	Data
Test Assurance Activity	The evaluator will acquire or develop an application that uses the selected OS mechanism with an X.509v3 certificate. The evaluator will then run the application and ensure that the provided certificate is used to authenticate the connection. The evaluator will repeat the activity for all selections listed.
Pass/Fail with Explanation	Pass. In test case FCS_TLSC_EXT.1.1 Test#1, The evaluator was able to monitor network traffic while the OS performs communication with user-initiated Application (Openssl s_client). Also, in test case FCS_TLSC_EXT.2.1 test #1 and 2, the evaluator was able to monitor network traffic while the OS performs communication with user-initiated Application (Openssl s_client), and update server. In all the above-mentioned test cases, the evaluator ensured that for each TLS and HTTPS connection a trusted channel was established and authenticated using an x509 certificate. This satisfies the testing requirement.

9.7 FTA

9.7.1 FTA_TAB.1

Item	Data
Test Assurance Activity	The evaluator will configure the OS, per instructions in the OS manual, to display the advisory warning message "TEST TEST Warning Message TEST TEST". The evaluator will then log out and confirm that the advisory message is displayed before logging in can occur.
Test Steps	1. Set up a new advisory warning message. 2. Ensure that the advisory message appears before the login process. 3. Validate that the administrator action for configuring the banner has been logged in the event log repository.

Expected Test Results	The TOE should allow the addition of an advisory message and display the message prior to login.
Pass/Fail with Explanation	Pass. The TOE allows the addition of an advisory message and displays the message prior to login. This satisfies the testing requirement.

9.8 FTP

9.8.1 FTP_ITC_EXT.1 (TD0789)

Item	Data
Test Assurance Activity	The evaluator shall configure the OS to communicate with another trusted IT product as identified in the third selection. The evaluator shall monitor network traffic while the OS performs communication with each of the servers identified in the third selection. The evaluator shall ensure that for each session a trusted channel was established in conformance with the selected protocols.
Pass/Fail with Explanation	Pass. In test case FCS_TLSC_EXT.1.1 Test#1, The evaluator was able to monitor network traffic while the OS performs communication with user-initiated Application (Openssl s_client). Also, in test case FCS_TLSC_EXT.2.1 test #1 and 2, the evaluator was able to monitor network traffic while the OS performs communication with user-initiated Application (Openssl s_client), and update server. In all the above-mentioned test cases, the evaluator ensured that for each session a trusted channel was established in conformance with TLS protocol as conforming to the Functional Package for Transport Layer Security (TLS), version 1.1. This satisfies the testing requirement.

9.8.2 FTP_TRP.1 Test#78 (TD0839)

Item	Data
Test Assurance Activity	The evaluator will also perform the following tests: • Test 78: The evaluator will ensure that communications using each remote or local administration method is tested during the course of the evaluation, setting up the connections or initial user authentication as described in the operational guidance and ensuring that communication is successful.
Test Steps	1. Verify TOE can be accessed via local console. 2. Login to the TOE and run the command “w” to list the logged in users 3. Verify the logs that connection is established.
Expected Test Results	Administrator should be able to access the TOE locally and verify that the communication was successful.
Pass/Fail with Explanation	Pass. The administrator can access the TOE using local administration method only since remote administration method is not selected in FTP_TRP1. This satisfies the testing requirement.

9.8.3 FTP_TRP.1 Test#79 (TD0839)

Item	Data
Test Assurance Activity	The evaluator will also perform the following tests:

	• Test 79: (Conditional: if “remote” is selected in FTP_TRP1.1). For each method of remote administration supported, the evaluator will follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote administrative session without invoking the trusted path
Pass/Fail with Explanation	N/A, because “remote” is not selected in FTP_TRP1.1.

9.8.4 FTP_TRP.1 Test#80 (TD0839)

Item	**Data**
Test Assurance Activity	The evaluator will also perform the following tests: • Test 80: (Conditional: if “remote” is selected in FTP_TRP1.1). The evaluator will ensure, for each method of remote administration, the channel data is not sent in plaintext.
Pass/Fail with Explanation	N/A, because “remote” is not selected in FTP_TRP1.1.

9.8.5 FTP_TRP.1 Test#81 (TD0839)

Item	**Data**
Test Assurance Activity	The evaluator will also perform the following tests: • Test 81: (Conditional: if “remote” is selected in FTP_TRP1.1). The evaluator will ensure, for each method of remote administration, modification of the channel data is detected by the OS.
Pass/Fail with Explanation	N/A, because “remote” is not selected in FTP_TRP1.1.

10 Conclusion

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

End of Document