Cisco Systems, Inc.

Cisco Catalyst 9800 Series Wireless Controllers and Access Points 17.6

Assurance Activity Report

Version 1.2

March 2023

Document prepared by

www.lightshipsec.com

Table of Contents

1 Introduction

1 This Assurance Activity Report (AAR) documents the evaluation activities performed by Lightship Security for the evaluation identified in Table 1. The AAR is produced in accordance with National Information Assurance Program (NIAP) reporting guidelines.

1.1 Evaluation Identifiers

Table 1: Evaluation Identifiers

Scheme	Canadian Common Criteria Scheme
Evaluation Facility	Lightship Security
Developer/Sponsor	Cisco Systems, Inc.
TOE	Cisco Catalyst 9800 Series Wireless Controllers and Access Points 17.6.01
Security Target	Cisco Catalyst 9800 Series Wireless Controllers and Access Points 17.6 Security Target, version 1.7, March 17, 2023
Protection Profile	collaborative Protection Profile for Network Devices, v2.2E (NDcPP), 23-March-2020 Network Device Collaborative Protection Profile (NDcPP) Extended Package Wireless Local Area Network (WLAN) Access Systems, May 29, 2015, Version 1.0

1.2 Evaluation Methods

2 The evaluation was performed using the methods, tools and standards identified in Table 2.

Table 2: Evaluation Methods

Evaluation Criteria	CC v3.1R5
Evaluation Methodology	CEM v3.1R5
Supporting Documents	Evaluation Activities for Network Device cPP, v2.2 (NDcPP-SD)
Interpretations	**NDcPP v2.2e+20200323** TD 0527 – Updates to Certificate Revocation Testing (FIA_X509_EXT.1) ***This TD applies to the TOE and AAs will be adhered to.*** TD 0528 - NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4 ***This TD does not apply to the TOE as it does not claim synchronising time with a NTP server (FCS_NTP_EXT.1).***

	TD 0536 - NIT Technical Decision for Update Verification Inconsistency ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0537 - NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3 ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0538 - NIT Technical Decision for Outdated link to allowed-with list ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0546 - NIT Technical Decision for DTLS – Clarification of Application Note 63 ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0547 - NIT Technical Decision for Clarification on Developer Disclosure of AVA_VAN ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0555 - NIT Technical Decision for RFC Reference Incorrect in TLSS Test ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0556 – NIT Technical Decision for RFC 5077 question ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0563 - NiT Technical Decision for Clarification of audit date information ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0564 - NiT Technical Decision for Vulnerability Analysis Search Criteria ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0569 - NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7 ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0570 - NiT Technical Decision for Clarification about FIA_AFL.1 ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0571 - NiT Technical Decision for Guidance on how to handle FIA_AFL.1 ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0572 - NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers ***This TD applies to the TOE and AAs will be adhered to.***

	TD 0580 - NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0581 - NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3 ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0591 - NIT Technical Decision for Virtual TOEs and hypervisors ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0592 - NIT Technical Decision for Local Storage of Audit Records ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0631 - NIT Technical Decision for Clarification of public key authentication for SSH Server ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0632 - NIT Technical Decision for Consistency with Time Data for vNDs ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0633 - NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0634 - NIT Technical Decision for Clarification required for testing IPv6 ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0635 - NIT Technical Decision for TLS Server and Key Agreement Parameters ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0636 - NIT Technical Decision for Clarification of Public Key User Authentication for SSH **This TD does not apply to the TOE as it does not claim SSH Client.**
	TD 0638 - NIT Technical Decision for Key Pair Generation for Authentication ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0639 - NIT Technical Decision for Clarification for NTP MAC Keys ***This TD applies to the TOE and AAs will be adhered to.***
	TD0670 - NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing

	This TD applies to the TOE and AAs will be adhered to.
	PP_WLAN_AS_EP_V1.0 20150529
	TD 0271 – RADsec as alternative to IPsec ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0282 – Test Activities added for Key Distribution and Key Generation ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0315 – Clarification of test for FCS_CKM.2.1(3) ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0456 – Removal of Low-level Crypto Failure Audit in WLAN AS EP ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0559 - Modes for AES Data Encryption/Decryption ***This TD applies to the TOE and AAs will be adhered to.***
	TD 0566 - Pre-Shared Keys ***This TD applies to the TOE and AAs will be adhered to.***
Tools	Please refer to the test plans.

1.3 Reference Documents

Table 3: List of Reference Documents

Ref	Document
[ST]	Cisco Catalyst 9800 Series Wireless Controllers and Access Points 17.6 Security Target, version 1.7, March 17, 2023
[AGD]	Cisco Catalyst 9800 Series Wireless Controllers and Access Points 17.6 CC Configuration Guide, version 0.8, February 10, 2023
[EST_REF]	Cisco libEST CC Testing Guide, v0.1
[SW_REF]	Cisco Catalyst 9800 Series Wireless Controller Software Configuration Guide, Cisco IOS XE Bengaluru 17.6.x https://www.cisco.com/c/en/us/td/docs/wireless/controller/9800/17-6/config-guide/b_wl_17_6_cg.html
[CMD_REF]	Cisco Catalyst 9800 Series Wireless Controller Command Reference, Cisco IOS XE Bengaluru 17.6.x https://www.cisco.com/c/en/us/td/docs/wireless/controller/9800/17-6/cmd-ref/b_wl_17_6_cr.html
[SEC_REF]	Cisco Catalyst 9800 Series Wireless Controller Software Configuration Guide, Cisco IOS XE Bengaluru 17.6.x

Ref	Document
	https://www.cisco.com/c/en/us/td/docs/wireless/controller/9800/17-6/config-guide/b_wl_17_6_cg/m_config_secure_shell_ewlc.html
[PP]	collaborative Protection Profile for Network Devices, v2.2E (NDcPP), 23-March-2020
[WLAN-EP]	Network Device Collaborative Protection Profile (NDcPP) Extended Package Wireless Local Area Network (WLAN) Access Systems, May 29, 2015, Version 1.0
[SD]	Evaluation Activities for Network Device cPP, v2.2 (NDcPP-SD)

2 Evaluation Activities for NDcPP SFRs

2.1 Security Audit (FAU)

2.1.1 FAU_GEN.1 Audit data generation

2.1.1.1 TSS

3 For the administrative task of generating/import of, changing, or deleting of cryptographic keys as defined in FAU_GEN.1.1c, the TSS should identify what information is logged to identify the relevant key.

Findings:	[ST] / TOE Summary Specification states, “When generating or deleting a cryptographic key the TOE will record an audit event in the audit log indicating the key with its associated label that was generated or deleted from key storage.”

4 For distributed TOEs the evaluator shall examine the TSS to ensure that it describes which of the overall required auditable events defined in FAU_GEN.1.1 are generated and recorded by which TOE components. The evaluator shall ensure that this mapping of audit events to TOE components accounts for, and is consistent with, information provided in Table 1, as well as events in Tables 2, 4, and 5 (where applicable to the overall TOE). This includes that the evaluator shall confirm that all components defined as generating audit information for a particular SFR should also contribute to that SFR as defined in the mapping of SFRs to TOE components, and that the audit records generated by each component cover all the SFRs that it implements.

Findings:	[ST] / TOE Summary Specification states, “A mapping is provided in table 25 to show which auditable events are covered by which components of the TOE.” “Each event is specified in the audit log enough detail to identify the user for which the event is associated, when the event occurred, where the event occurred, the outcome of the event, and the type of event that occurred. “ The evaluator confirmed that the mapping of audit events to TOE components in [ST] Table 25 accounts for and is consistent with information provided in [PP] Tables 1, 2, 4 and 5 (where applicable to the overall TOE) and in [WLAN-EP] Table 1. The evaluator compared Table 21 and Table 25 in the [ST] and confirmed that all components defined as generating audit information for a particular SFR should also contribute to that SFR, and that the audit records generated by each component cover all the SFRs that it implements.

2.1.1.2 Guidance Documentation

5 The evaluator shall check the guidance documentation and ensure that it provides an example of each auditable event required by FAU_GEN.1 (i.e. at least one instance of each auditable event, comprising the mandatory, optional and selection-based SFR sections as applicable, shall be provided from the actual audit record).

Findings:	Tables 8 and 9 in the Auditing section of [AGD] provide examples of each auditable event as required by FAU_GEN.1 for each mandatory, optional and selection-based SFR of all claimed protection profiles and extended packages.

6 The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes. The evaluator shall examine the guidance documentation and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are related to TSF data related to configuration changes. The evaluator may perform this activity as part of the activities associated with ensuring that the corresponding guidance documentation satisfies the requirements related to it.

Findings:	The evaluator performed this activity as a part of those Assurance Activities associated with ensuring the corresponding guidance documentation satisfies their independent requirements. Overall, the evaluator considered the administrator guides published by the vendor. The evaluator reviewed the contents of these documents and looked specifically for functionality related to the scope of the evaluation.

2.1.1.3 Tests

7 The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the table of audit events and administrative actions listed above. This should include all instances of an event: for instance, if there are several different I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the guidance documentation, and that the fields in each audit record have the proper entries.

8 For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of auditable events to TOE components in the Security Target. For all events involving more than one TOE component when an audit event is triggered, the evaluator has to check that the event has been audited on both sides (e.g. failure of building up a secure communication channel between the two components). This is not limited to error cases but includes also events about successful actions like successful build up/tear down of a secure communication channel between TOE components.

9 Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.

Findings:	These tests are conducted throughout the test plan. In the evaluated configuration, the TOE acts as a centralized point for collecting and forwarding audit information from TOE components. As such, auditable events associated with a TOE component are immediately forwarded to the TOE and logged. Tests associated with auditable events of TOE components are covered by the above tests. Also note that some testing activities must be conducted prior to TOE component registration/DTLS secure channel establishment. In such cases, TOE components are configured to log directly to their serial console.

2.1.2 FAU_GEN.2 User identity association

2.1.2.1 TSS & Guidance Documentation

10 The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1.

2.1.2.2 Tests

11 This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

12 For distributed TOEs the evaluator shall verify that where auditable events are instigated by another component, the component that records the event associates the event with the identity of the instigator. The evaluator shall perform at least one test on one component where another component instigates an auditable event. The evaluator shall verify that the event is recorded by the component as expected and the event is associated with the instigating component. It is assumed that an event instigated by another component can at least be generated for building up a secure channel between two TOE components. If for some reason (could be e.g. TSS or Guidance Documentation) the evaluator would come to the conclusion that the overall TOE does not generate any events instigated by other components, then this requirement shall be omitted.

Findings:	These activities are performed in conjunction with the testing of FAU_GEN.1.1.

2.1.3 FAU_STG_EXT.1 Protected audit event storage

2.1.3.1 TSS

13 The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.

Findings:	[ST] / TOE Summary Specification states, “The WLC, which is the component that stores audit data locally, will transmit all audit messages in real-time to a specified, external syslog server. The WLC protects communications with an external syslog server using IPsec.”

14 The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access.

Findings:	[ST] / TOE Summary Specification states, “If the IPsec connection inadvertently fails, the TOE will buffer between 4096-bytes and 2,148,483,647 bytes of audit records on the TOE. When connectivity with its configured syslog server is restored, the WLC will transmit the buffer contents. The exact size of the audit storage is configured using the “logging buffered” command. If the local logging limit is reached, the oldest messages overwritten to accommodate the new message.” “The WLC protects the local logging buffer from unauthorized access, modification or deletion. No account is able to modify data that has been written to the local logging buffer. Only the Administrator is able to clear the local logging buffer.”

15 The evaluator shall examine the TSS to ensure it describes whether the TOE is a standalone TOE that stores audit data locally or a distributed TOE that stores audit data locally on each TOE component or a distributed TOE that contains TOE components that cannot store audit data locally on themselves but need to transfer audit data to other TOE components that can store audit data locally. The evaluator

shall examine the TSS to ensure that for distributed TOEs it contains a list of TOE components that store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs that contain components which do not store audit data locally but transmit their generated audit data to other components it contains a mapping between the transmitting and storing TOE components.

Findings: [ST] / TOE Summary Specification states, "The Cisco Catalyst 9800 Series Wireless Controllers and Access Points 17.6 TOE is distributed. After the AP joins the WLC to form a distributed TOE the AP will transmit its audit messages to the WLC over the secure DTLS channel described in FPT_ITT.1. A mapping between the transmitting and storing TOE components is provided below.

Transmitting Component	Storing Component
AP	WLC

"

16 The evaluator shall examine the TSS to ensure that it details the behaviour of the TOE when the storage space for audit data is full. When the option 'overwrite previous audit record' is selected this description should include an outline of the rule for overwriting audit data. If 'other actions' are chosen such as sending the new audit data to an external IT entity, then the related behaviour of the TOE shall also be detailed in the TSS.

Findings: [ST] / TOE Summary Specification states, "The exact size of the audit storage is configured using the "logging buffered" command. If the local logging limit is reached, the oldest messages overwritten to accommodate the new message."

17 The evaluator shall examine the TSS to ensure that it details whether the transmission of audit information to an external IT entity can be done in real-time or periodically. In case the TOE does not perform transmission in real-time the evaluator needs to verify that the TSS provides details about what event stimulates the transmission to be made as well as the possible as well as acceptable frequency for the transfer of audit data.

Findings: [ST] / TOE Summary Specification states, "The WLC, which is the component that stores audit data locally, will transmit all audit messages in real-time to a specified, external syslog server."

18 For distributed TOEs the evaluator shall examine the TSS to ensure it describes to which TOE components this SFR applies and how audit data transfer to the external audit server is implemented among the different TOE components (e.g. every TOE components does its own transfer or the data is sent to another TOE component for central transfer of all audit events to the external audit server).

Findings: [ST] / TOE Summary Specification identifies that this SFR applies to both the Wireless LAN Controller (WLC) and the Access Point (AP).
[ST] states, "After the AP joins the WLC to form a distributed TOE the AP will transmit its audit messages to the WLC over the secure DTLS channel described in FPT_ITT.1."
"The WLC, which is the component that stores audit data locally, will transmit all audit messages in real-time to a specified, external syslog server."

19 For distributed TOEs the evaluator shall examine the TSS to ensure it describes which TOE components are storing audit information locally and which components are buffering audit information and forwarding the information to another TOE component for local storage. For every component the TSS shall describe the behaviour when local storage space or buffer space is exhausted.

Findings:	[ST] /TOE Summary Specification (FAU_STG_EXT.4 FAU_STG_EXT.5) states, "The AP maintains the audit data in a transmission buffer and continues to do so until the AP has transferred its contents to the WLC where it is stored locally." For the WLC "If the local logging limit is reached, the oldest messages overwritten to accommodate the new message." For the AP "Under normal operating conditions the AP transmission buffer will never become exhausted. Should an unlikely event occur where the transmission buffer becomes exhausted, the oldest message in the buffer will be overwritten to accommodate the new message."

2.1.3.2 Guidance Documentation

20 The evaluator shall also examine the guidance documentation to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server.

Findings:	The "Procedures and Operational Guidance for IT Environment" section of the [AGD] states "Syslog Server. Any syslog server that can be accessed over IPsec may be used." Instructions on how to configure IPsec and Syslog can be found in the "IPsec" subsection of the section "Preparative Procedures and Operational Guidance for the TOE" [AGD].

21 The evaluator shall also examine the guidance documentation to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.

Findings:	The "Enable Remote Syslog Sever" subsection of the "IPsec" section of the [AGD] states the following, "When an audit event is generated, is it simultaneously sent to the external server and the local store."

22 The evaluator shall also ensure that the guidance documentation describes all possible configuration options for FAU_STG_EXT.1.3 and the resulting behaviour of the TOE for each possible configuration. The description of possible configuration options and resulting behaviour shall correspond to those described in the TSS.

Findings:	[AGD] section "Configure Local Logging Buffer Size" provides configuration options for local logging buffer size in a range of 4096 to 2,148,483,647 bytes. The [AGD] states, " If the local storage space for audit data is full the TOE will overwrite the oldest audit record to make room for the new audit record."

2.1.3.3 Tests

23 Testing of the trusted channel mechanism for audit will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall perform the following additional tests for this requirement:

a) Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe

that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing. The evaluator shall verify that the TOE is capable of transferring audit data to an external audit server automatically without administrator intervention.

High-Level Test Description
Verification that the data is encrypted is satisfied by FTP_ITC.1 for the log forwarding channel. The logging server is a syslog-ng v3.8.1 which receives log data over IPsec enabled by strongSwan version U5.5.1/K4.9.0-13-amd64 as described in the Test Setup. Due to the log-forwarding mechanism used on logging server, the audit records are therefore confirmed to have been successfully received by the audit server whenever the test cases are run.
Findings: PASS

b) Test 2: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.3. Depending on the configuration this means that the evaluator has to check the content of the audit data when the audit data is just filled to the maximum and then verifies that

1) The audit data remains unchanged with every new auditable event that should be tracked but that the audit data is recorded again after the local storage for audit data is cleared (for the option ‘drop new audit data’ in FAU_STG_EXT.1.3).

2) The existing audit data is overwritten with every new auditable event that should be tracked according to the specified rule (for the option ‘overwrite previous audit records’ in FAU_STG_EXT.1.3)

3) The TOE behaves as specified (for the option ‘other action’ in FAU_STG_EXT.1.3).

High-Level Test Description
Adjust local logging buffer size to facilitate test. Show the local logging buffer. Generate additional events and review the local logging buffer again. Confirm that older events are dropped off.
Findings: PASS

c) Test 3: If the TOE complies with FAU_STG_EXT.2/LocSpace the evaluator shall verify that the numbers provided by the TOE according to the selection for FAU_STG_EXT.2/LocSpace are correct when performing the tests for FAU_STG_EXT.1.3

Test Not Applicable:	The TOE does not claim this functionality.

d) Test 4: For distributed TOEs, Test 1 defined above should be applicable to all TOE components that forward audit data to an external audit server. For the local storage according to FAU_STG_EXT.1.2 and FAU_STG_EXT.1.3 the Test 2 specified above shall be applied to all TOE components that store audit data locally. For all TOE components that store audit data locally and comply with FAU_STG_EXT.2/LocSpace Test 3 specified above shall be applied. The evaluator shall verify that the transfer of audit data to an external audit server is implemented.

High-Level Test Description
Verification that TOE component audit data is encrypted is satisfied by FTP_ITC.1 and FCS_DTLSS_EXT.1/FCS_DTLSC_EXT.1. In the evaluated configuration, TOE components do not store audit data locally.
Findings: PASS

2.2 Cryptographic Support (FCS)

2.2.1 FCS_CKM.1/KeyGen Cryptographic Key Generation

2.2.1.1 TSS

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

Findings:	[ST] / TOE Summary Specification identifies the key sizes supported by the TOE and the usage of each scheme. The evaluator confirmed this information is consistent with the rest of the ST.

The following table describes the key generation algorithms the TOE implements to generate asymmetric keys used for **device authentication**:

Scheme	Standard	Key Size/ NIST Curve	SFR	Service
RSA	FIPS PUB 186-4	2048 3072	FCS_SSHS_EXT.1	SSH Remote Administration
			FCS_TLSC_EXT.1 /RADsec FCS_TLSC_EXT.2	RADsec
			FCS_TLSC_EXT.1/EST FCS_TLSC_EXT.2	EST Server
			FCS_TLSS_EXT.1	HTTPS Remote Administration
			FCS_DTLSS_EXT.1 FCS_DTLSS_EXT.2	DTLS Server
			FCS_IPSEC_EXT.1	Transmit generated audit data to an external IT entity
ECC	FIPS PUB 186-4	P-256 P-384	FCS_TLSS_EXT.1	HTTPS Remote Administration
			FCS_IPSEC_EXT.1	Transmit generated audit data to an external IT entity
			FCS_TLSC_EXT.1/EST FCS_TLSC_EXT.2	EST Server
ECC	FIPS PUB 186-4	P-384	FCS_DTLSS_EXT.1 FCS_DTLSS_EXT.2	DTLS Server
			FCS_DTLSC_EXT.1 FCS_DTLSC_EXT.2	DTLS Client

Scheme	Standard	Key Size/ NIST Curve	SFR	Service
RSA	FIPS PUB 186-4	2048	FCS_TLSC_EXT.2	RADsec
ECC	FIPS PUB 186-4	P-256 P-384	FCS_TLSS_EXT.1	HTTPS Remote Administration
			FCS_IPSEC_EXT.1	Transmit generated audit data to an external IT entity
			FCS_SSHS_EXT.1	SSH Remote Administration
			FCS_TLSC_EXT.1/EST FCS_TLSC_EXT.2	EST Server
ECC	FIPS PUB 186-4	P-384	FCS_DTLSS_EXT.1 FCS_DTLSS_EXT.2	DTLS Server
			FCS_DTLSC_EXT.1 FCS_DTLSC_EXT.2	DTLS Client
FFC	FIPS PUB 186-4	2048	FCS_DTLSS_EXT.1 FCS_DTLSS_EXT.2	DTLS Server
			FCS_DTLSC_EXT.1 FCS_DTLSC_EXT.2	DTLS Client
			FCS_TLSC_EXT.1/EST FCS_TLSC_EXT.2	EST Server

2.2.1.2 Guidance Documentation

25 The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all cryptographic protocols defined in the Security Target.

Findings: The [ST] claims key generation for RSA, ECC (P-256, P-384 and P-521) and FFC (group 19 and 20) schemes. Instructions on how to configure the key generation scheme and key size for a given TSF are provided in the associated section of the [AGD], namely, SSH, HTTPS, IPsec, TLS-RADsec, DTLS-CAPWAP and CC Mode.

In all cases, the administrator specifies the key generation scheme upon generation of new keys using one of the following commands, “crypto key generate ec keysize

<key size> label <key name>…]" and "crypto key generate rsa label <key name> modulus <key size>…".

Additionally, the [ST] claims symmetric key generation for WPA2 using PRF-384 and PRF-704. Instructions on how to configure WPA2 key generation schemes and key sizes are found in the Configure WLANs section of the [AGD].

2.2.1.3 Tests

26 Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products. Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported).

Key Generation for FIPS PUB 186-4 RSA Schemes

27 The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent *e*, the private prime factors *p* and *q*, the public modulus *n* and the calculation of the private signature exponent *d*.

28 Key Pair generation specifies 5 ways (or methods) to generate the primes *p* and *q*. These include:

a. Random Primes:

- Provable primes
- Probable primes

b. 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

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

Findings: See table below for CAVP mapping.

TOE Component	Cryptographic operation	NIST Standard	SFR(s) supported	CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)	CAVP certificate number

IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	RSA Key Generation	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3;	FCS_DTLSC_EXT.1	RSA KeyGen (FIPS186-4)	A2452
Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	RSA Key Generation	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3;	FCS_DTLSC_EXT.1	RSA KeyGen (FIPS186-4)	A877
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	RSA Key Generation	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3;	FCS_TLSS_EXT.1 FCS_TLSC_EXT.1 FCS_DTLSS_EXT.1 FCS_SSHS_EXT.1 FCS_IPSEC_EXT.1	RSA KeyGen (FIPS186-4)	A2452 A1462

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

CAVP A1462 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13937

Key Generation for Elliptic Curve Cryptography (ECC)

FIPS 186-4 ECC Key Generation Test

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

FIPS 186-4 Public Key Verification (PKV) Test

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

Findings:	See table below for CAVP mapping.

TOE Component	**Cryptographic operation**	**NIST Standard**	**SFR(s) supported**	**CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)**	**CAVP certificate number**
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	ECDSA Key Generation	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	FCS_DTLSC_EXT.1	ECDSA KeyGen (FIPS186-4)	A2452
Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	ECDSA Key Generation	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	FCS_DTLSC_EXT.1	ECDSA KeyGen (FIPS186-4)	A877
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	ECDSA Key Generation	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	FCS_TLSS_EXT.1 FCS_TLSC_EXT.1 FCS_DTLSS_EXT.1 FCS_SSHS_EXT.1 FCS_IPSEC_EXT.1	ECDSA KeyGen (FIPS186-4)	A2452 A1462

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

CAVP A1462 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13937

Key Generation for Finite-Field Cryptography (FFC)

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

33 The Parameter generation specifies 2 ways (or methods) to generate the cryptographic prime q and the field prime p:

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

34 and two ways to generate the cryptographic group generator g:

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

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

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

36 The security strength of the RBG must be at least that of the security offered by the FFC parameter set.

37 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.

38 For each key length supported, the evaluator shall have the TSF generate 25 parameter sets and key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. Verification must also confirm

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

39 for each FFC parameter set and key pair.

Findings:	See table below for CAVP mapping.

TOE Component	**Cryptographic operation**	**NIST Standard**	**SFR(s) supported**	**CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)**	**CAVP certificate number**
IW6300 ESW6300 AP 1562 AP 2802 AP 3802	FFC Key generation of size 2048 bits or greater.	FIPS PUB 186-4, "Digital Signature Standard	FCS_DTLSC_EXT.1	DSA KeyGen (FIPS186-4)	A2452

AP 4800		(DSS)", Appendix B.1			
Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	FFC Key generation of size 2048 bits or greater.	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1	FCS_DTLSC_EXT.1	DSA KeyGen (FIPS186-4)	A877
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	FFC Key generation of size 2048 bits or greater.	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1	FCS_DTLSS_EXT.1 FCS_TLSC_EXT.1	DSA KeyGen (FIPS186-4)	A2452

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

NOTE:	Modified per TD0580.

FFC Schemes using "safe-prime" groups

40 Testing for FFC Schemes using safe-prime groups is done as part of testing in CKM.2.1.

Findings:	Done as part of testing in CKM.2.1.

2.2.2 FCS_CKM.2/KeyEst Cryptographic Key Establishment

2.2.2.1 TSS

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

Findings:	The evaluator confirmed that the supported key establishment schemes in FCS_CKM.2.1/KeyEst correspond to the key generation schemes identified in FCS_CKM.1.1/KeyGen.

[ST] / TOE Summary Specification identifies the usage for each scheme.

The following table shows the methods the TOE implements for **key establishment**:

Scheme	Standard	SFR	Service
RSAES-PKCS1-v1_5	Section 7.2 of RFC 3447	FCS_TLSC_EXT.2	RADsec
		FCS_TLSS_EXT.1	HTTPS Remote Administration
EC-DH	NIST SP 800-56A Revision 2	FCS_TLSS_EXT.1	HTTPS Remote Administration
		FCS_IPSEC_EXT.1	Transmit generated audit data to an external IT entity
		FCS_SSHS_EXT.1	SSH Remote Administration
		FCS_TLSC_EXT.1 FCS_TLSC_EXT.2	EST Server
		FCS_DTLSS_EXT.1 FCS_DTLSS_EXT.2	DTLS Server
		FCS_DTLSC_EXT.1 FCS_DTLSC_EXT.2	DTLS Client
FFC	NIST SP 800-56A Revision 2	FCS_DTLSS_EXT.1 FCS_DTLSS_EXT.2	DTLS Server
		FCS_DTLSC_EXT.1 FCS_DTLSC_EXT.2	DTLS Client
		FCS_TLSC_EXT.1 FCS_TLSC_EXT.2	EST Server

NOTE:	Removed per TD0580.

42 ~~If Diffie-Hellman group 14 is selected from FCS_CKM.2.1, the TSS shall claim the TOE meets RFC 3526 Section 3.~~

43 The intent of this activity is to be able to identify the scheme being used by each service. This would mean, for example, one way to document scheme usage could be:

NOTE:	Removed per TD0580.

Scheme	SFR	Service
RSA	FCS_TLSS_EXT.1	Administration
ECDH	FCS_SSHC_EXT.1	Audit Server
~~Diffie-Hellman (Group 14)~~	~~FCS_SSHC_EXT.1~~	~~Backup Server~~
ECDH	FCS_IPSEC_EXT.1	Authentication Server

44 The information provided in the example above does not necessarily have to be included as a table but can be presented in other ways as long as the necessary data is available.

2.2.2.2 Guidance Documentation

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

Findings:	The [ST] claims key establishment for RSA, ECC (P-256, P-384 and P-521) and FFC (group 19 and 20) schemes. Instructions on how to configure the key establishment schemes for a given TSF are provided in the associated section of the [AGD], namely, SSH, HTTPS, IPsec, TLS-RADsec, DTLS-CAPWAP and CC Mode. For SSH, ECDH key establishment schemes may be configured using the following command, “ip ssh server algorithm kex <ecdh kex method>” For HTTPS, RSA or ECDH key establishment schemes can be configured using the following command, “ip http secure-ciphersuite <ciphersuites>”. For IPsec, DH key establishment schemes can be configured using the following command, “group <19 | 20>” in the ikev2 proposal sub-menu. For TLS-RADsec, RSA key establishment scheme is used by default and requires no specific configuration. For DTLS-CAPWAP, DH and ECDH key establishment schemes can be configured using the following command, “ap dtls-ciphersuite priority <priority> <ciphersuite>”. For the TLS connection used by the TOE to connect to the EST server, DH or ECDH key establishment schemes are used by default, according to which certificate types the EST server trustpoints use. No additional configuration is necessary.

2.2.2.3 Tests

Key Establishment Schemes

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

SP800-56A Key Establishment Schemes

47 The evaluator shall verify a TOE's implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for

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

Function Test

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

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

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

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

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

Validity Test

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

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

55 The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOE's results

with the results using a known good implementation verifying that the TOE detects these errors.

Findings:	See table below for CAVP mapping.

TOE Component	**Cryptographic operation**	**NIST Standard**	**SFR(s) supported**	**CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)**	**CAVP certificate number**
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	Elliptic curve-based key establishment	NIST Special Publication 800-56A Revision 2, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;	FCS_DTLSC_EXT.1	KAS-ECC-SSC Sp800-56Ar3	A2452
Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	Elliptic curve-based key establishment	NIST Special Publication 800-56A Revision 2, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;	FCS_DTLSC_EXT.1	KAS-ECC-SSC Sp800-56Ar3	A877
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	Elliptic curve-based key establishment	NIST Special Publication 800-56A Revision 2, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;	FCS_TLSS_EXT.1 FCS_DTLSS_EXT.1 FCS_SSHS_EXT.1 FCS_IPSEC_EXT.1	KAS-ECC-SSC Sp800-56Ar3	A2452 A1462
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	Finite field-based key establishment		FCS_DTLSC_EXT.1	KAS-FFC-SSC Sp800-56Ar3	A2452

Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	Finite field-based key establishment		FCS_DTLSC_EXT.1	KAS-FFC-SSC Sp800-56Ar3	A877
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	Finite field-based key establishment		FCS_DTLSS_EXT.1 FCS_TLSC_EXT.1 FCS_TLSS_EXT.1	KAS-FFC-SSC Sp800-56Ar3	A2452

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

CAVP A1462 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13937

RSA-based key establishment schemes

56 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_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses RSAES-PKCS1-v1_5.

Findings: See table below for CAVP mapping.

TOE Component	**Cryptographic operation**	**NIST Standard**	**SFR(s) supported**	**CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)**	**CAVP certificate number**
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	RSA-based Key Establishment	RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards	FCS_TLSS_EXT.1 FCS_TLSC_EXT.1	No CAVP exists, must be described in TSS. -SHS Validation List	A2452

		(PKCS) #1: RSA Cryptography Specifications Version 2.1"		(**SHA-1, SHA2-256, SHA2-384**) - Hash algorithms as applicable DRBG Validation List (**Counter DRBG**) - Supported Random Bit Generators (DRBG) RSA Validation List - An RSA key pair generation algorithm in FIPS 186-4 (**See RSA/KeyGen in FCS_CKM.1**)	

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

NOTE:	Removed per TD0580.

~~Diffie-Hellman Group 14~~

57 ~~The evaluator shall verify the correctness of the TSF's implementation of Diffie-Hellman group 14 by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses Diffie-Hellman group 14.~~

FFC Schemes using "safe-prime" groups

58 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_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses safe-prime groups. This test must be performed for each safe-prime group that each protocol uses.

Findings:	Safe primes were tested as part of the FCS_IPSEC_EXT.1 protocol testing using a known good implementation (Strongswan).

2.2.3 FCS_CKM.4 Cryptographic Key Destruction

2.2.3.1 TSS

59 The evaluator examines the TSS to ensure it lists all relevant keys (describing the origin and storage location of each), all relevant key destruction situations (e.g. factory reset or device wipe function, disconnection of trusted channels, key change as part of a secure channel protocol), and the destruction method used in each case. For the purpose of this Evaluation Activity the relevant keys are those keys that are relied upon to support any of the SFRs in the Security Target. The evaluator confirms that the description of keys and storage locations is consistent with the functions carried out by the TOE (e.g. that all keys for the TOE-specific secure channels and protocols, or that support FPT_APW.EXT.1 and FPT_SKP_EXT.1, are accounted for[1]). In particular, if a TOE claims not to store plaintext keys in non-volatile memory then the evaluator checks that this is consistent with the operation of the TOE.

Findings: [ST] / Table 22 lists all relevant keys with the origin and storage locations, all relevant key destruction situations, and the destruction method used in each case. Below is a portion of table 22 for example.

Key	Description	Storage Location	Zeroization Method
HTTPS TLS Encryption Key	HTTPS TLS Encryption Key	SDRAM	Overwritten automatically with 0x00 when the HTTPS session is no longer in use.

60 The evaluator shall check to ensure the TSS identifies how the TOE destroys keys stored as plaintext in non-volatile memory, and that the description includes identification and description of the interfaces that the TOE uses to destroy keys (e.g., file system APIs, key store APIs).

Findings: [ST] / Table 22 identifies how the TOE destroys keys stored in plaintext in non-volatile memory; they are “Overwritten with 0x00 by using the following command: #crypto key zeroize <label>”.

61 Note that where selections involve ‘*destruction of reference*’ (for volatile memory) or ‘*invocation of an interface’* (for non-volatile memory) then the relevant interface definition is examined by the evaluator to ensure that the interface supports the selection(s) and description in the TSS. In the case of non-volatile memory the evaluator includes in their examination the relevant interface description for each media type on which plaintext keys are stored. The presence of OS-level and storage device-level swap and cache files is not examined in the current version of the Evaluation Activity.

Findings: [ST] / Table 22 describes the mechanism by which the TOE destroys plaintext keys in non-volatile memory.

62 Where the TSS identifies keys that are stored in a non-plaintext form, the evaluator shall check that the TSS identifies the encryption method and the key-encrypting-key

[1] Where keys are stored encrypted or wrapped under another key then this may need to be explained in order to allow the evaluator to confirm the consistency of the description of keys with the TOE functions.

used, and that the key-encrypting-key is either itself stored in an encrypted form or that it is destroyed by a method included under FCS_CKM.4.

Findings:	The TSS does not identify any keys that are stored in a non-plaintext form. [ST] / TOE Summary Specification (FPT_SKP_EXT.1) states, "The TOE is designed specifically to not disclose any keys stored in the TOE. The TOE stores all private keys in a secure directory that cannot be viewed or accessed, even by the Administrator. The TOE stores symmetric keys only in volatile memory."

63 The evaluator shall check that the TSS identifies any configurations or circumstances that may not conform to the key destruction requirement (see further discussion in the Guidance Documentation section below). Note that reference may be made to the Guidance Documentation for description of the detail of such cases where destruction may be prevented or delayed.

Findings:	The TSS does not identify any configuration or circumstances that may not conform to the key destruction requirement.

64 Where the ST specifies the use of "a value that does not contain any CSP" to overwrite keys, the evaluator examines the TSS to ensure that it describes how that pattern is obtained and used, and that this justifies the claim that the pattern does not contain any CSPs.

Findings:	The ST does not claim this selection.

2.2.3.2 Guidance Documentation

65 A TOE may be subject to situations that could prevent or delay key destruction in some cases. The evaluator shall 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 supporting information used). The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer.

66 For example, when the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-levelling and garbage collection. This may result in additional copies of the key that are logically inaccessible but persist physically. Where available, the TOE might then describe use of the TRIM command[2] and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

Findings:	The [AGD] does not identify any situations where key destruction is delayed or prevented. The evaluator confirmed this is consistent with the TSS.

2.2.3.3 Tests

67 None

[2] Where TRIM is used then the TSS and/or guidance documentation is also expected to describe how the keys are stored such that they are not inaccessible to TRIM, (e.g. they would need not to be contained in a file less than 982 bytes which would be completely contained in the master file table).

2.2.4 FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption)

2.2.4.1 TSS

68 The evaluator shall examine the TSS to ensure it identifies the key size(s) and mode(s) supported by the TOE for data encryption/decryption.

Findings:	[ST] / TOE Summary Specification identifies that the TOE supports AES with the following modes: CBC, GCM, CCMP, GCMP and CTR, and the key sizes 128 bit and 256 bit.

2.2.4.2 Guidance Documentation

69 The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected mode(s) and key size(s) defined in the Security Target supported by the TOE for data encryption/decryption.

Findings:	The [ST] claims AES in CBC, CCMP, CTR, GCM, and GCMP modes with cryptographic key sizes of 128 and 256 bits. Instructions on how to configure encryption modes and key sizes for a given TSF are provided in the associated section of the [AGD], namely, SSH, HTTPS, IPsec, TLS-RADsec, DTLS-CAPWAP and CC Mode. For SSH, AES encryption algorithms may be configured using the following command, “ip ssh server algorithm encryption <encryption algorithm>”. For HTTPS, AES encryption algorithms can be configured using the following command, “ip http secure-ciphersuite <ciphersuites>”. For IPsec, AES encryption algorithms can be configured using the following command, “encryption <aes-gcm-128 | aes-gcm-256>” in the ikev2 proposal sub-menu. Note these only apply to the ESP. Encryption of IKEv2 payloads is not configurable and the TOE uses AES-GCM-[128,256] by default. For TLS-RADsec, encryption algorithms are not configurable and AES-128-CBC is used by default. For DTLS-CAPWAP, encryption algorithms can be configured using the following command, “ap dtls-ciphersuite priority <priority> <ciphersuite>”. For the TLS connection used by the TOE to connect to the EST server, AES-128 or AES-256 encryption algorithms are used by default. No additional configuration is necessary

2.2.4.3 Tests

AES-CBC Known Answer Tests

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

71 **KAT-1.** To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all-zeros key.

72 To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption.

73 **KAT-2.** To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys.

74 To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using an all-zero ciphertext value as input and AES-CBC decryption.

75 **KAT-3.** To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key *i* in each set shall have the leftmost *i* bits be ones and the rightmost *N-i* bits be zeros, for *i* in [1,N].

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

77 **KAT-4.** To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256-bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128].

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

AES-CBC Multi-Block Message Test

79 The evaluator shall test the encrypt functionality by encrypting an *i*-block message where 1 < *i* <=10. The evaluator shall choose a key, an IV and plaintext message of length *i* blocks and encrypt the message, using the mode to be tested, with the chosen

key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation.

80 The evaluator shall also test the decrypt functionality for each mode by decrypting an *i*-block message where 1 < *i* <=10. The evaluator shall choose a key, an IV and a ciphertext message of length *i* blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.

AES-CBC Monte Carlo Tests

81 The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:

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

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

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

Findings:	See table below for CAVP mapping.

TOE Component	**Cryptographic operation**	**NIST Standard**	**SFR(s) supported**	**CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)**	**CAVP certificate number**
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	AES-CBC	AES-CBC	FCS_DTLSC_EXT.1	AES-CBC	A2452
Catalyst 9130 Catalyst 9115	AES-CBC	AES-CBC	FCS_DTLSC_EXT.1	AES-CBC	A877

Catalyst 9120 Catalyst 9105					
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	AES-CBC	AES-CBC	FCS_TLSS_EXT.1 FCS_TLSC_EXT.1 FCS_DTLSS_EXT.1 FCS_SSHS_EXT.1	AES-CBC	A2452 A1462

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

CAVP A1462 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13937

AES-GCM Test

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

128 bit and 256 bit keys

a. **Two plaintext lengths**. One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported.

a. **Three AAD lengths**. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported.

b. **Two IV lengths**. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.

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

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

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

Findings:	See table below for CAVP mapping.

TOE Component	**Cryptographic operation**	**NIST Standard**	**SFR(s) supported**	**CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)**	**CAVP certificate number**
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	AES-GCM	AES-GCM	FCS_DTLSC_EXT.1	AES-GCM	A2452
Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	AES-GCM	AES-GCM	FCS_DTLSC_EXT.1	AES-GCM	A877
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	AES-GCM	AES-GCM	FCS_TLSS_EXT.1 FCS_TLSC_EXT.1 FCS_DTLSS_EXT.1	AES-GCM	A2452 A1462
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800 Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	AES-GCMP	AES-GCMP-256 (as defined in NIST SP800-38D and IEEE 802.11ac-2013)	FCS_COP.1/DataEncryption	AES-GCM	A2452 A877 See Table 24 in the [ST] for Wifi Alliance certificates for IEEE 802.11ac-2013.

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

CAVP A1462 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13937

AES-CTR Known Answer Tests

88 The Counter (CTR) mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. Since the Counter Mode does not specify the counter that is used, it is not possible to implement an automated test for this mode. The generation and management of the counter is tested through FCS_SSH*_EXT.1.4. If CBC and/or GCM are selected in FCS_COP.1/DataEncryption, the test activities for those modes sufficiently demonstrate the correctness of the AES algorithm. If CTR is the only selection in FCS_COP.1/DataEncryption, the AES-CBC Known Answer Test, AES-GCM Known Answer Test, or the following test shall be performed (all of these tests demonstrate the correctness of the AES algorithm):

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

90 KAT-1 To test the encrypt functionality, the evaluator shall supply a set of 5 plaintext values for each selected keysize and obtain the ciphertext value that results from encryption of the given plaintext using a key value of all zeros.

91 KAT-2 To test the encrypt functionality, the evaluator shall supply a set of 5 key values for each selected keysize and obtain the ciphertext value that results from encryption of an all zeros plaintext using the given key value.

92 KAT-3 To test the encrypt functionality, the evaluator shall supply a set of key values for each selected keysize as described below and obtain the ciphertext values that result from AES encryption of an all zeros plaintext using the given key values. A set of 128 128-bit keys, a set of 192 192-bit keys, and/or a set of 256 256-bit keys. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1, N].

93 KAT-4 To test the encrypt functionality, the evaluator shall supply the set of 128 plaintext values described below and obtain the ciphertext values that result from encryption of the given plaintext using each selected keysize with a key value of all zeros (e.g. 256 ciphertext values will be generated if 128 bits and 256 bits are

selected and 384 ciphertext values will be generated if all keysizes are selected). Plaintext value i in each set shall have the leftmost bits be ones and the rightmost 128-i bits be zeros, for i in [1, 128]

AES-CTR Multi-Block Message Test

94 The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10 (test shall be performed using AES-ECB mode). For each i the evaluator shall choose a key 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 using a known good implementation. The evaluator shall perform this test using each selected keysize.

AES-CTR Monte-Carlo Test

95 The evaluator shall test the encrypt functionality using 100 plaintext/key pairs. The plaintext values shall be 128-bit blocks. For each pair, 1000 iterations shall be run as follows:

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

96 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 evaluator shall perform this test using each selected keysize.

Findings:	The TOE does not claim AES-CTR cryptographic algorithms.

[Updated per testing requirements in PP WLAN AS EP v1.0]

AES-CCM Tests

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

128 bit and 256 bit keys

Two payload lengths. One payload length shall be the shortest supported payload length, greater than or equal to zero bytes. The other payload length shall 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 shall be 0, if supported. One associated data length shall be the shortest supported payload length, greater than or equal to zero bytes. One associated data length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits). If the implementation supports an associated data length of 216 bytes, an associated data length of 216 bytes shall be tested.

Nonce lengths. All supported nonce lengths between 7 and 13 bytes, inclusive, shall be tested.

Tag lengths. All supported tag lengths of 4, 6, 8, 10, 12, 14 and 16 bytes shall be tested.

98 Due to the restrictions that IEEE 802.11 specifies for this mode (nonce length of 13 and tag length of 8), it is acceptable to test a subset of the supported lengths as long as the selections fall into the ranges specified above. In this case, the evaluator shall ensure that these are the only supported lengths.

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

100 Test 1. For EACH supported key and associated data length and ANY supported payload, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

101 Test 2. For EACH supported key and payload length and ANY supported associated data, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

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

103 Test 4. For EACH supported key and tag length and ANY supported associated data, payload and nonce length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

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

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

106 Additionally, the evaluator shall 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 AES-CCMP Encapsulation Example and Section 2.2 Additional AES CCMP Test Vectors to further verify the IEEE 802.11-2012 implementation of AES-CCMP.

Findings:	See table below for CAVP mapping.

TOE Component	**Cryptographic operation**	**NIST Standard**	**SFR(s) supported**	**CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)**	**CAVP certificate number**
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	AES-CCMP	AES-CCMP (as defined in NIST SP 800-	FCS_COP.1/DataEncryption	AES-CCM	A2452 A877 See Table 24 in the

Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105		38C and IEEE 802.11-2012)			[ST] for Wifi Alliance certificates.
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	AES-CCMP-128 AES-CCMP-256 AES-GCMP-128 AES-GCMP-256	AES-CCMP (as defined in NIST SP 800-38C and IEEE 802.11-2012) AES-GCMP (as specified in NIST SP800-38D and IEEE 802.11ac-2013)	FCS_COP.1/DataEncryption	AES-CCM-128 AES-CCM-256 AES-GCM-128 AES-GCM-256	AES 4114 See Table 24 in the [ST] for Wifi Alliance certificates.
Catalyst 9130	AES-CCMP-128 AES-CCMP-256 AES-GCMP-128 AES-GCMP-256	AES-CCMP (as defined in NIST SP 800-38C and IEEE 802.11-2012) AES-GCMP (as specified in NIST SP800-38D and IEEE 802.11ac-2013)	FCS_COP.1/DataEncryption	AES-CCM-128 AES-CCM-256 AES-GCM-128 AES-GCM-256	AES 5663 See Table 24 in the [ST] for Wifi Alliance certificates.
Catalyst 9115 Catalyst 9120	AES-CCMP-128 AES-GCMP-128	AES-CCMP (as defined in NIST	FCS_COP.1/DataEncryption	AES-CCM-128 AES-GCM-128	C1273 See Table 24 in the [ST] for Wifi

		SP 800-38C and IEEE 802.11-2012) AES-GCMP (as specified in NIST SP800-38D and IEEE 802.11ac-2013)			Alliance certificates
Catalyst 9105	AES-CCMP-128 AES-GCMP-128	AES-CCMP (as defined in NIST SP 800-38C and IEEE 802.11-2012) AES-GCMP (as specified in NIST SP800-38D and IEEE 802.11ac-2013)	FCS_COP.1/DataEncryption	AES-CCM-128 AES-GCM-128	C1275 See Table 24 in the [ST] for Wifi Alliance certificates

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

CAVP 1273 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?validation=31679

CAVP 1275 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?validation=31681

AES 5663 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?validation=21654

AES 4114 - https://csrc.nist.gov/Projects/Cryptographic-Algorithm-Validation-Program/details?source=AES&number=4114

2.2.5 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

2.2.5.1 TSS

107 The evaluator shall examine the TSS to determine that it specifies the cryptographic algorithm and key size supported by the TOE for signature services.

Findings:	[ST] / TOE Summary Specification states, “The TOE provides cryptographic signature services using Elliptic Curve Digital Signature Algorithm with a key size of 256 and 384 bits and RSA Digital Signature Algorithm with key size of 2048 and greater, as specified in FIPS PUB 186-4, “Digital Signature Standard.””

2.2.5.2 Guidance Documentation

108 The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected cryptographic algorithm and key size defined in the Security Target supported by the TOE for signature services.

Findings:	The [ST] claims RSA and ECDSA schemes for signature services. Instructions on how to configure encryption modes and key sizes for a given TSF are provided in the associated section of the [AGD], namely, SSH, HTTPS, IPsec, TLS-RADsec, DTLS-CAPWAP and CC Mode. In all cases, the administrator specifies the key type upon generation of new keys using one of the following commands, “crypto key generate ec keysize <key size> label <key name>…]” and “crypto key generate rsa label <key name> modulus <key size>…”. Such keys can be used for ECDSA and RSA signature services respectively.

2.2.5.3 Tests

ECDSA Algorithm Tests

ECDSA FIPS 186-4 Signature Generation Test

109 For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation.

ECDSA FIPS 186-4 Signature Verification Test

110 For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.

Findings:	See table below for CAVP mapping.

TOE Component	Cryptographic operation	NIST Standard	SFR(s) supported	CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)	CAVP certificate number
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	ECDSA Signature Generation and Verification	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384]; ISO/IEC 14888-3, Section 6.4	FCS_DTLSC_EXT.1	ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)	A2452
Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	ECDSA Signature Generation and Verification	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384]; ISO/IEC 14888-3, Section 6.4	FCS_DTLSC_EXT.1	ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)	A877
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	ECDSA Signature Generation and Verification	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384]; ISO/IEC 14888-3, Section 6.4	FCS_TLSS_EXT.1 FCS_TLSC_EXT.1 FCS_DTLSS_EXT.1 FCS_IPSEC_EXT.1	ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4)	A2452 A1462

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

CAVP A1462 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13937

RSA Signature Algorithm Tests

Signature Generation Test

111 The evaluator generates or obtains 10 messages for each modulus size/SHA combination supported by the TOE. The TOE generates and returns the corresponding signatures.

112 The evaluator shall verify the correctness of the TOE's signature using a trusted reference implementation of the signature verification algorithm and the associated public keys to verify the signatures.

Signature Verification Test

113 For each modulus size/hash algorithm selected, the evaluator generates a modulus and three associated key pairs, (*d*, *e*). Each private key *d* is used to sign six pseudorandom messages each of 1024 bits using a trusted reference implementation of the signature generation algorithm. Some of the public keys, *e*, messages, or signatures are altered so that signature verification should fail. For both the set of original messages and the set of altered messages: the modulus, hash algorithm, public key *e* values, messages, and signatures are forwarded to the TOE, which then attempts to verify the signatures and returns the verification results.

114 The evaluator verifies that the TOE confirms correct signatures on the original messages and detects the errors introduced in the altered messages.

Findings: See table below for CAVP mapping.

TOE Component	**Cryptographic operation**	**NIST Standard**	**SFR(s) supported**	**CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)**	**CAVP certificate number**
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	RSA Signature Generation and Verification	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital	FCS_DTLSC_EXT.1	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	A2452

		Signature scheme 3			
Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	RSA Signature Generation and Verification	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	FCS_DTLSC_EXT.1	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	A877
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	RSA Signature Generation and Verification	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	FCS_TLSS_EXT.1 FCS_TLSC_EXT.1 FCS_DTLSS_EXT.1 FCS_SSHS_EXT.1 FCS_IPSEC_EXT.1	RSA SigGen (FIPS186-4) RSA SigVer (FIPS186-4)	A2452 A1462

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

CAVP A1462 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13937

2.2.6 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

2.2.6.1 TSS

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

Findings:	[ST] / TOE Summary Specification describes the hash functions and the associations with other TSF cryptographic functions. ■ SSH – SHA-1, SHA-256, and SHA-512 ■ HTTPS (TLS Server) – SHA-1, SHA-256, and SHA-384 ■ TLS Client (RADsec) – SHA-1 ■ DTLS Server (CAPWAP) – SHA-2 ■ IKE/IPSEC – SHA-1, SHA-256, SHA-384, and SHA-512 ■ Image Verification and Software Integrity - SHA-512

2.2.6.2 Guidance Documentation

116 The evaluator checks the AGD documents to determine that any configuration that is required to configure the required hash sizes is present.

Findings:	The [ST] claims SHA-1, SHA-256, SHA-384 and SHA-512. Instructions on how to configure hash sizes for a given TSF are provided in the associated section of the [AGD], namely, SSH, HTTPS, IPsec, TLS-RADsec, DTLS-CAPWAP and CC Mode. For SSH, hash size may be configured using the following command, “ip ssh server algorithm hostkey rsa-sha2-256 rsa-sha2-512”. For HTTPS, hash size can be configured using the following command, “ip http secure-ciphersuite <ciphersuites>”. For IPsec, no additional configuration is required to ensure the required hash sizes are present. For TLS-RADsec, no additional configuration is required to ensure the required hash sizes are present. For DTLS-CAPWAP, hash size can be configured using the following command, “ap dtls-ciphersuite priority <priority> <ciphersuite>”. For the TLS connection used by the TOE to connect to the EST server, no additional configuration is required to ensure the required hash sizes are present.

2.2.6.3 Tests

117 The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF 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 testmacs.

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

Short Messages Test - Bit-oriented Mode

119 The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Short Messages Test - Byte-oriented Mode

120 The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Selected Long Messages Test - Bit-oriented Mode

121 The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Selected Long Messages Test - Byte-oriented Mode

122 The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Pseudorandomly Generated Messages Test

123 This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

Findings: See table below for CAVP mapping.

TOE Component	Cryptographic operation	NIST Standard	SFR(s) supported	CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)	CAVP certificate number
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	SHA-1, SHA2-256, SHA2-384)	*ISO/IEC 10118-3:2004.*	FCS_DTLSC_EXT.1	SHA-1, SHA2-256, SHA2-384)	A2452

Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	SHA-1, SHA2-256, SHA2-384)	*ISO/IEC 10118-3:2004.*	FCS_DTLSC_EXT.1	SHA-1, SHA2-256, SHA2-384)	A877
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	SHA-1, SHA2-256, SHA2-384, SHA2-512)	*ISO/IEC 10118-3:2004.*	FCS_TLSS_EXT.1 FCS_TLSC_EXT.1 FCS_DTLSS_EXT.1 FCS_SSHS_EXT.1 FCS_IPSEC_EXT.1	SHA-1, SHA2-256, SHA2-384, SHA2-512)	A2452 A1462

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

CAVP A1462 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13937

2.2.7 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

2.2.7.1 TSS

124 The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used.

Findings:	[ST] / TOE Summary specification states, “The TOE provides keyed-hashing message authentication services using HMAC-SHA-1 and HMAC-SHA-256 that operates on 512-bit blocks and HMAC-SHA-384 and HMAC-SHA-512 operating on 1024-bit blocks of data, with key sizes and message digest sizes of 160-bits, 256 bits, 384 bits, and 512 bits respectively as specified in ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.“

2.2.7.2 Guidance Documentation

125 The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the values used by the HMAC function: key length, hash function used, block size, and output MAC length used defined in the Security Target supported by the TOE for keyed hash function.

Findings:	The [ST] claims support for HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512 with key sizes of 160, 256, 384, 512 bits and message digest sizes of 160, 256, 384, 512 bits. Instructions on how to configure HMAC for a given TSF are provided in the associated section of the [AGD], namely, SSH. To configure HMAC for SSH, the following command may be used: "ip ssh server algorithm mac hmac-sha2-512 hmac-sha2-256".

2.2.7.3 Tests

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

Findings:	See table below for CAVP mapping.

TOE Component	**Cryptographic operation**	**NIST Standard**	**SFR(s) supported**	**CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)**	**CAVP certificate number**
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-384	*ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*	FCS_DTLSC_EXT.1	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384	A2452
Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-384	*ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*	FCS_DTLSC_EXT.1	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384	A877
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	*ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*	FCS_TLSS_EXT.1 FCS_TLSC_EXT.1 FCS_DTLSS_EXT.1 FCS_SSHS_EXT.1 FCS_IPSEC_EXT.1	HMAC-SHA-1, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512	A2452 A1462

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

CAVP A1462 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13937

2.2.8 FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

127 Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Appendix D of [PP].

2.2.8.1 TSS

128 The evaluator shall examine the TSS to determine that it specifies the DRBG type, identifies the entropy source(s) seeding the DRBG, and state the assumed or calculated min-entropy supplied either separately by each source or the min-entropy contained in the combined seed value.

Findings:	[ST] / TOE Summary Specification states, “The TSF implements a random bit generator (RBG) based on the AES-256 block cipher, in accordance with ISO/IEC 18031:2011. This DRBG is seeded with a hardware entropy source that provides 256 bits of entropy to the DRBG.”

2.2.8.2 Guidance Documentation

129 The evaluator shall confirm that the guidance documentation contains appropriate instructions for configuring the RNG functionality.

Findings:	The [ST] claims CTR_DRBG (AES) for random number generation functionality. The [AGD] does not identify any configurable RNG functionality.

2.2.8.3 Tests

130 The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration.

131 If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP800-90A).

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

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

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

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

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

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

Findings: See table below for CAVP mapping.

TOE Component	**Cryptographic operation**	**NIST Standard**	**SFR(s) supported**	**CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)**	**CAVP certificate number**
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	CTR DRBG (AES)	*ISO/IEC 18031:2011 using [CTR_DRBG (AES)]*	FCS_DTLSC_EXT.1	Counter DRBG	A2452
Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	CTR DRBG (AES)	*ISO/IEC 18031:2011 using [CTR_DRBG (AES)]*	FCS_DTLSC_EXT.1	Counter DRBG	A877
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	CTR DRBG (AES)	*ISO/IEC 18031:2011 using [CTR_DRBG (AES)]*	FCS_TLSS_EXT.1 FCS_TLSC_EXT.1 FCS_DTLSS_EXT.1 FCS_SSHS_EXT.1 FCS_IPSEC_EXT.1	Counter DRBG	A2452 A1462

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

CAVP A1462 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13937

2.3 Identification and Authentication (FIA)

2.3.1 FIA_AFL.1 Authentication Failure Management

2.3.1.1 TSS

134 The evaluator shall examine the TSS to determine that it contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked. The TSS shall also describe the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability.

Findings:	[ST] / TOE Summary Specification states, “To block password-based brute force attacks, the TOE uses an internal AAA function to detect and track failed login attempts. When an account attempting to log into an administrative interface reaches the set maximum number of failed authentication attempts, the account will not be granted access until the time period has elapsed or until the Administrator manually unblocks the account.“

135 The evaluator shall examine the TSS to confirm that the TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking).

Findings:	[ST] / TOE Summary Specification states, “To avoid a potential situation where password failures made by Administrators leads to no Administrator access until the defined blocking time period has elapsed, the CC Configuration Guide instructs the Administrator to configure the Controller for SSH public key access, which is not subjected to password-based brute force attacks.“

2.3.1.2 Guidance Documentation

136 The evaluator shall examine the guidance documentation to ensure that instructions for configuring the number of successive unsuccessful authentication attempts and time period (if implemented) are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each “action” specified (if that option is chosen). If different actions or mechanisms are implemented depending on the secure protocol employed (e.g., TLS vs. SSH), all must be described.

Findings:	The Configure Authentication Failure subsection of the Preparative Procedures and Operational Guidance for the TOE section of the [AGD] provides instructions on configuring the number of successive unsuccessful login attempts in a specified time period resulting in user lockout for a specified duration. The configuration applies to all password-based authentication mechanisms which includes the HTTPS web GUI and SSH CLI. As described in the section, this is accomplished using the following command, “aaa authentication rejected <1-25> in <1-65535> ban <1-65535>”.

137 The evaluator shall examine the guidance documentation to confirm that it describes, and identifies the importance of, any actions that are required in order to ensure that administrator access will always be maintained, even if remote administration is made permanently or temporarily unavailable due to blocking of accounts as a result of FIA_AFL.1.

Findings:	The Unblock Locked-Out Account subsection of the section, Operational Guidance for the TOE, in the [AGD] states the following, “To unblock an account that has been prevented from logging in due to successive login failures enter the following: WLC# clear aaa local user blocked username <username>” Furthermore, the SSH section of the [AGD] states, “During the defined lockout period, the Controller provides the ability for the Administrator account to login remotely using SSH public key authentication.”

2.3.1.3 Tests

138 The evaluator shall perform the following tests for each method by which remote administrators access the TOE (e.g. any passwords entered as part of establishing the connection protocol or the remote administrator application):

a. Test 1: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE (and, if the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall also use the operational guidance to configure the time period after which access is re-enabled). The evaluator shall test that once the authentication attempts limit is reached, authentication attempts with valid credentials are no longer successful.

High-Level Test Description
Using the TSFI, set the lockout threshold to be 3 failed attempts within 3 minutes for a lockout duration of 5 minutes.
Using the CLI SSH interface, log into the TOE twice using an incorrect password. On the third attempt, log in correctly and verify that the threshold has not been reached.
Using the CLI SSH interface, log into the TOE three times using an incorrect password. On the fourth attempt, log in correctly and verify that the threshold has been reached and that the user cannot log in. Then, wait for the remainder of the lockout duration and verify the user can now log in.
Using the CLI SSH interface, log into the TOE three times using an incorrect password. On the fourth attempt, log in correctly and verify that the threshold has been reached and that the user cannot log in.
Manually unlock the user using the TSFI and verify the user’s lockout status has been reset.

High-Level Test Description
Verify the user can now log in. Repeat the above tests using the web GUI login.
Findings: PASS

b. Test 2: After reaching the limit for unsuccessful authentication attempts as in Test 1 above, the evaluator shall proceed as follows.

If the administrator action selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall confirm by testing that following the operational guidance and performing each action specified in the ST to re-enable the remote administrator's access results in successful access (when using valid credentials for that administrator).

Findings:	See previous test case.

If the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall wait for just less than the time period configured in Test 1 and show that an authorisation attempt using valid credentials does not result in successful access. The evaluator shall then wait until just after the time period configured in Test 1 and show that an authorisation attempt using valid credentials results in successful access.

Findings:	See previous test case.

2.3.2 FIA_PMG_EXT.1 Password Management

2.3.2.1 TSS

139 The evaluator shall examine the TSS to determine that it contains the lists of the supported special character(s) and minimum and maximum number of characters supported for administrator passwords.

Findings:	[ST] / TOE Summary Specification states, "The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")" and other special characters listed in table 18. Minimum password length is settable by the Authorized Administrator, and can be configured for minimum password lengths of 8 to 16 characters and maximum of 127 characters."

2.3.2.2 Guidance Documentation

140 The evaluator shall examine the guidance documentation to determine that it:

a. identifies the characters that may be used in passwords and provides guidance to Security Administrators on the composition of strong passwords, and

b. provides instructions on setting the minimum password length and describes the valid minimum password lengths supported.

Findings:	The [AGD] provides instructions on configuring minimum password lengths and the special characters that may be used in passwords in the Physical Controller — Initial Configuration, Virtual Controller — Initial Configuration, Define Password Policy and Add Administrator Account subsections of the section, Preparative Procedures and Operational Guidance for the TOE. **The** Virtual Controller — Initial Configuration and Physical Controller — Initial Configuration subsections of the section, Preparative Procedures and Operational Guidance for the TOE state, “Compose a password with a length between 8 and 16 using any combination of upper and lower case letters, numbers, and the following special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)” Table 4 of the Add Administrator Account subsection of the section, Preparative Procedures and Operational Guidance for the TOE provides a detailed list of all supported special characters. The Define Password Policy subsection of the section, Preparative Procedures and Operational Guidance for the TOE describes how a password policy with minimum password length can be configured using the “aaa common-criteria policy <policy name>” and “min-length <8-16>” commands.

2.3.2.3 Tests

141 The evaluator shall perform the following tests.

a. Test 1: The evaluator shall compose passwords that meet the requirements in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, and a minimum length listed in the requirement are supported and justify the subset of those characters chosen for testing.

High-Level Test Description
Change the minimum password length to be 15 characters using a common-criteria password policy. Associate the policy with the testadmin account and simultaneously change the account password. Show that the password can be used to login to the TOE. Change the password for the user to a password which is less than the configured minimum and show it is rejected. Change the global minimum password length to be 15 characters. Change the password for the current admin to a password which is less than the configured minimum and show it is rejected. Change the password the admin to be 15 characters and show it is accepted.
Findings: PASS

b. Test 2: The evaluator shall compose passwords that do not meet the requirements in some way. For each password, the evaluator shall verify that the TOE does not support the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that the TOE enforces the allowed characters and the minimum length listed in the requirement and justify the subset of those characters chosen for testing.

Findings:	See previous test case.

2.3.3 FIA_UIA_EXT.1 User Identification and Authentication

2.3.3.1 TSS

142 The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon".

Findings:	TOE Summary Specification states, "The requirement applies to users of the Controllers who connect locally to the CLI via serial console or over SSH and HTTPS remote administrative interfaces." "Administrative access to the TOE is facilitated through a local password-based authentication and SSH public key authentication mechanisms on the Controller through which all Administrator actions are mediated. Once a potential (unauthenticated) administrative user attempts to access the TOE through an interactive administrative interface (CLI or GUI), the TOE prompts the user for a user name and password or SSH public key authentication. No access is allowed to the administrative functionality of the TOE until the administrator is successfully identified and authenticated." "After the end-user provides a username and authentication credentials the TOE grants administrative access (if credentials are valid, and the account has not been locked) or indicates that the login attempt was unsuccessful. At the CLI a successful login is indicated by a hash sign ("#") next to the device hostname. At the HTTPS Web GUI, a successful login is indicated by providing the Administrator with the default landing page, which is the Wireless Dashboard."

143 The evaluator shall examine the TSS to determine that it describes which actions are allowed before user identification and authentication. The description shall cover authentication and identification for local and remote TOE administration.

Findings:	[ST] / TOE Summary Specification states, "The WLC component requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed." "There are no local or remote management administrative interfaces directly available on the Access Points. Additionally, there are no unauthenticated services provided or supported. All administration of the APs are performed via the WLC."

144 For distributed TOEs the evaluator shall examine that the TSS details how Security Administrators are authenticated and identified by all TOE components. If not all TOE components support authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2, the TSS shall describe how the overall TOE functionality is split between TOE components including how it is ensured that no unauthorized access to any TOE component can occur.

Findings:	[ST] / TOE Summary Specification states, "The WLC component requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed" "Administrative access to the TOE is facilitated through a local password-based authentication and SSH public key authentication mechanisms".

	“All administration of the APs are performed via the WLC. If an attempt is made to directly connect to the local serial port of the AP, it will respond with the following message: “Console disabled while in FIPS mode”.”

145 For distributed TOEs, the evaluator shall examine the TSS to determine that it describes for each TOE component which actions are allowed before user identification and authentication. The description shall cover authentication and identification for local and remote TOE administration. For each TOE component that does not support authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2 the TSS shall describe any unauthenticated services/services that are supported by the component.

Findings:	[ST] / TOE Summary Specification sates, **“**The WLC component requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed.” “Prior to authentication at each interactive administrative interfaces (CLI and GUI), the TOE may be configured by the Administrator to display a customized login banner” “There are no local or remote management administrative interfaces directly available on the Access Points. Additionally, there are no unauthenticated services provided or supported”

2.3.3.2 Guidance Documentation

146 The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps (e.g., establishing credential material such as pre-shared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the guidance documentation provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the guidance documentation provides sufficient instruction on limiting the allowed services.

Findings:	Instructions on how to configure administrator login through the HTTPS, local and SSH interfaces is provided in the HTTPS, Configure Local Authentication, and SSH sections respectively. Prior to being able to login via HTTPS, the administrator must follow the instructions in the HTTPS section. The HTTP server can then be started using the command, “ip http server”. Prior to being able to login via SSH, the administrator must follow the instructions in the SSH section. The SSH server can then be enabled using the command “transport input ssh” in the line configuration sub-menu. The [AGD] does not identify any configurations necessary to enable logging in via the local console. The [AGD] does not identify any configurations required to limit the services provided prior to login as there are no such services.

2.3.3.3 Tests

147 The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method:

a. Test 1: The evaluator shall use the guidance documentation to configure the appropriate credential supported for the login method. For that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.

High-Level Test Description
Log into the identified management interface using a known-good credential and logout. Attempt to log into the identified management interface using a known-bad credential. Ensure the appropriate audit messages appear.
Findings: PASS

b. Test 2: The evaluator shall configure the services allowed (if any) according to the guidance documentation, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.

High-Level Test Description
The device does not have any services configured prior to I&A. All claimed services available to remote entities are identified as part of AVA_VAN.1 test scanning.
Findings: PASS

c. Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.

High-Level Test Description
The device does not have any services configured prior to I&A. All claimed services available to local entities are identified as part of AVA_VAN.1 test scanning.
Findings: PASS

d. Test 4: For distributed TOEs where not all TOE components support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2, the evaluator shall test that the components authenticate Security Administrators as described in the TSS.

Findings:	TOE components do not support authentication of Security Administrators in the evaluated configuration.

2.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

148 Evaluation Activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1.

2.3.5 FIA_UAU.7 Protected Authentication Feedback

2.3.5.1 TSS

149 None

2.3.5.2 Guidance Documentation

150 The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps to ensure authentication data is not revealed while entering for each local login allowed.

Findings:	The [AGD] does not identify any necessary steps to ensure authentication data is not revealed. Password characters are not echoed back to the user at the local CLI by default.

2.3.5.3 Tests

151 The evaluator shall perform the following test for each method of local login allowed:

a. Test 1: The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.

High-Level Test Description
Log into the local management interface. Ensure the password field does not echo any characters to the display, as claimed by the ST.
Findings: PASS

2.4 Security management (FMT)

2.4.1 General requirements for distributed TOEs

2.4.1.1 TSS

152 For distributed TOEs it is required to verify the TSS to ensure that it describes how every function related to security management is realized for every TOE component and shared between different TOE components. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs.

Findings:	[ST] / TOE Summary Specification includes the component that implements the security management functions. The evaluator confirmed that all relevant aspects of each TOE component are covered by the FMT SFRs.

2.4.1.2 Guidance Documentation

153 For distributed TOEs it is required to verify the Guidance Documentation to describe management of each TOE component. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs.

Findings:	The [AGD] provides adequate instructions to manage TOE components throughout the [AGD]. It should be noted that in the evaluated configuration, all TOE components (namely, access points) are centrally managed through the TOE. In this configuration, no management functionality is provided by TOE components directly.

2.4.1.3 Tests

154 Tests defined to verify the correct implementation of security management functions shall be performed for every TOE component. For security management functions that are implemented centrally, sampling should be applied when defining the evaluator's tests (ensuring that all components are covered by the sample).

Findings:	The TOE is distributed. In the evaluated configuration, all management functionality is implemented centrally and no interface is provided to directly interact with TOE components. As such, tests defined to verify the correct implementation of security management functions are applicable to all TOE components.

2.4.2 FMT_MOF.1/ManualUpdate

2.4.2.1 TSS

155 For distributed TOEs see [SD] chapter 2.4.1.1. There are no specific requirements for non-distributed TOEs.

Findings:	The evaluator confirmed that the Wireless LAN Controller (WLC) component is responsible for managing manual updates. [ST] / TOE Summary Specification (FMT_MOF.1/Services FMT_MOF.1/Functions FMT_MTD.1/CryptoKeys) states, "Only the authorized Administrator on the WLC may: ■ Initiate manual updates of the TOE software; …"

2.4.2.2 Guidance Documentation

156 The evaluator shall examine the guidance documentation to determine that any necessary steps to perform manual update are described. The guidance documentation shall also provide warnings regarding functions that may cease to operate during the update (if applicable).

Findings:	Necessary steps to perform manual update of the TOE and TOE components are provided in the Update WLC and AP Software subsection of the section, Operational Guidance for the TOE of the [AGD]. As described in the subsection, the WLC and AP software may be simultaneously updated using the command, "install add file [tftp | ftp | sftp://<IP Address of TFTP/FTP/SFTP server>] <image name.bin> activate commit"

	Furthermore, the subsection states, “The WLC will commit the new image, save the configuration, and reload. All APs that are joined to the WLC will automatically reboot when the WLC reboots.”

157 For distributed TOEs the guidance documentation shall describe all steps how to update all TOE components. This shall contain description of the order in which components need to be updated if the order is relevant to the update process. The guidance documentation shall also provide warnings regarding functions of TOE components and the overall TOE that may cease to operate during the update (if applicable).

Findings:	Necessary steps to perform manual update of the TOE and TOE components are provided in the Update WLC and AP Software subsection of the section, Operational Guidance for the TOE of the [AGD]. As described in the subsection, the WLC and AP software may be simultaneously updated using the command, “install add file [tftp | ftp | sftp://<IP Address of TFTP/FTP/SFTP server>] <image name.bin> activate commit” Furthermore, the subsection states, “The WLC will commit the new image, save the configuration, and reload. All APs that are joined to the WLC will automatically reboot when the WLC reboots.”

2.4.2.3 Tests

158 The evaluator shall try to perform the update using a legitimate update image without prior authentication as Security Administrator (either by authentication as a user with no administrator privileges or without user authentication at all – depending on the configuration of the TOE). The attempt to update the TOE shall fail.

High-Level Test Description
Log into the TOE using an account with privileges which should not permit upgrades. Attempt to upgrade the device. The action should fail. Repeat test using the Web GUI.
Findings: PASS

159 The evaluator shall try to perform the update with prior authentication as Security Administrator using a legitimate update image. This attempt should be successful. This test case should be covered by the tests for FPT_TUD_EXT.1 already.

Findings:	This test case is covered in FPT_TUD_EXT.1.

2.4.3 FMT_MTD.1/CoreData Management of TSF Data

2.4.3.1 TSS

160 The evaluator shall examine the TSS to determine that, for each administrative function identified in the guidance documentation; those that are accessible through an interface prior to administrator log-in are identified. For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

Findings:	[ST] / TOE Summary Specification states, "… all Admin functions including those functions that manage TSF data are mediated by the TOE which ensures there is no capability to manage TSF data at any administrative interface until an administrator is successfully identified and authenticated."

161 If TOE supports handling of X.509v3 certificates and implements a trust store, the evaluator shall examine the TSS to determine that it contains sufficient information to describe how the ability to manage the TOE's trust store is restricted.

Findings:	[ST] / TOE Summary Specification states, "In addition, the TOE ensures management of truststores (trustpoints) containing X.509 certificates is restricted to the authorized Administrator. User accounts with less than level 15 privilege do not have the ability to add or remove a truststore."

2.4.3.2 Guidance Documentation

162 The evaluator shall review the guidance documentation to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of the cPP is identified, and that configuration information is provided to ensure that only administrators have access to the functions.

Findings:	The [AGD] and its associated reference documents list all functions that can be used to manipulate TSF-data. By default, only privileged administrators have access to such functions (no configuration is necessary).

163 If the TOE supports handling of X.509v3 certificates and provides a trust store, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to configure and maintain the trust store in a secure way. If the TOE supports loading of CA certificates, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to securely load CA certificates into the trust store. The evaluator shall also review the guidance documentation to determine that it explains how to designate a CA certificate a trust anchor.

Findings:	Instructions on how to configure and maintain the trust store in a secure way, loading of CA certificates into the trust store and designating CA certificates as trust anchors are provided in sections of the [AGD] corresponding to TSFs relying on X.509v3 certificates/CAs, namely, the HTTPS, IPsec, TLS, DTLS, CC Mode subsections of the section, Preparative Procedures and Operational Guidance for the TOE and the Adding New APs subsections of the section, Operational Guidance for the TOE of the [AGD]. In all cases, the TOE trust store is managed using the command, "crypto pki trustpoint <trustpoint name>" and associated trustpoint sub-menu commands as well as the "crypto pki authenticate <trustpoint name>" command. These commands are used to create CA, intermediate CA certificates.

2.4.3.3 Tests

164 No separate testing for FMT_MTD.1/CoreData is required unless one of the management functions has not already been exercised under any other SFR.

2.4.4 FMT_SMF.1 Specification of Management Functions

165 The security management functions for FMT_SMF.1 are distributed throughout the cPP and are included as part of the requirements in FTA_SSL_EXT.1, FTA_SSL.3, FTA_TAB.1, FMT_MOF.1/ManualUpdate, FMT_MOF.1/AutoUpdate (if included in the ST), FIA_AFL.1, FIA_X509_EXT.2.2 (if included in the ST), FPT_TUD_EXT.1.2

& FPT_TUD_EXT.2.2 (if included in the ST and if they include an administrator-configurable action), FMT_MOF.1/Services, and FMT_MOF.1/Functions (for all of these SFRs that are included in the ST), FMT_MTD, FPT_TST_EXT, and any cryptographic management functions specified in the reference standards. Compliance to these requirements satisfies compliance with FMT_SMF.1.

2.4.4.1 TSS (containing also requirements on Guidance Documentation and Tests)

166 The evaluator shall examine the TSS, Guidance Documentation and the TOE as observed during all other testing and shall confirm that the management functions specified in FMT_SMF.1 are provided by the TOE. The evaluator shall confirm that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface).

Findings:	[ST] / TOE Summary Specification states, “The Administrator can connect to the WLC to perform management functions via a directly connected console cable. The Administrator can also connect (from wired networks) remotely to the WLC over TLS/HTTPS or SSH to perform management functions.” The TSS listed the security management functions supported by the TOE and identified that they were accessible on all interfaces to the WLC.

167 The evaluator shall examine the TSS and Guidance Documentation to verify they both describe the local administrative interface. The evaluator shall ensure the Guidance Documentation includes appropriate warnings for the administrator to ensure the interface is local.

Findings:	[ST] / TOE Summary Specification describes the local administrative interface.

168 For distributed TOEs with the option 'ability to configure the interaction between TOE components' the evaluator shall examine that the ways to configure the interaction between TOE components is detailed in the TSS and Guidance Documentation. The evaluator shall check that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS and Guidance Documentation.

Findings:	[ST] / TOE Summary Specification states, “Maintain an AP authorization list to allow the Administrator to configure the interaction between the WLC and APs and which APs are allowed to join. Refer to FCO_CPC_EXT.1 for further details.”

2.4.4.2 Guidance Documentation

169 See section 2.4.4.1.

2.4.4.3 Tests

170 The evaluator tests management functions as part of testing the SFRs identified in section 2.4.4. No separate testing for FMT_SMF.1 is required unless one of the management functions in FMT_SMF.1.1 has not already been exercised under any other SFR.

Findings:	During testing, the evaluator reviewed the claims in FMT_SMF.1.1, verified each explicitly declared management function was covered by at least one other SFR and that audit information for the management function was covered by the SFR test and included in the test findings. No other management function was identified during testing that have not already been exercised.

2.4.5 FMT_SMR.2 Restrictions on security roles

2.4.5.1 TSS

171 The evaluator shall examine the TSS to determine that it details the TOE supported roles and any restrictions of the roles involving administration of the TOE.

Findings:	[ST] / TOE Summary Specification states, “The Administrator is dependent upon having a level 15 privilege. A user without a level 15 privilege would not have the ability to enable, disable or modify security functional management behavior.”

2.4.5.2 Guidance Documentation

172 The evaluator shall review the guidance documentation to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration.

Findings:	The [AGD] describes the use of the CLI throughout. The respective Hardware Installation Guides identified in [AGD] Table 1 describe connecting to the local console port. The [AGD] describes remote administration via HTTPS and SSH in the Remote Administration Protocols subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD]. As described in the SSH subsection of the Remote Administration Protocols subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD], SSH remote administration can be enabled using the “transport input ssh” command from the virtual terminal sub menu. As described in the HTTPS subsection of the Remote Administration Protocols subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD], HTTPS remote administration can be enabled using the “ip http server” command. The [AGD] does not identify any necessary client configuration to enable SSH or HTTPS remote administration.

2.4.5.3 Tests

173 In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this cPP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team’s test activities.

Findings:	There are no explicit test activities and therefore none are recorded here. All interfaces are tested throughout this test plan.

2.5 Protection of the TSF (FPT)

2.5.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

2.5.1.1 TSS

174 The evaluator shall examine the TSS to determine that it details how any pre-shared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

Findings:	[ST] / TOE Summary Specification states, “The TOE is designed specifically to not disclose any keys stored in the TOE. The TOE stores all private keys in a secure directory that cannot be viewed or accessed, even by the Administrator. The TOE stores symmetric keys only in volatile memory. Pre-shared keys may be specified in the configuration file by the Administrator using a bit-based (hex) format Only the Administrator may view the configuration file.”

2.5.2 FPT_APW_EXT.1 Protection of Administrator Passwords

2.5.2.1 TSS

175 The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and the method used to obscure the plaintext password data when stored. The TSS shall also detail passwords are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note.

Findings:	[ST] / TOE Summary Specification states, “The TOE is designed specifically to not disclose any passwords stored in the TOE. All passwords are stored using a SHA-2 hash. ‘Show’ commands display only the hashed password.”

2.5.3 FPT_TST_EXT.1 TSF testing

2.5.3.1 TSS

176 The evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

Findings:	[ST] / TOE Summary Specification details the self-tests and what the tests are doing. Below is an example of one of the descriptions of one of the self-tests that are performed: “■ HMAC Known Answer Test - For each of the hash values listed, the HMAC implementation is fed known plaintext data and a known key. These values are used to generate a MAC. This MAC is compared to a known MAC to verify that the HMAC and hash operations are operating correctly.” “All TOE components (WLC and AP) will automatically verify the integrity of the

	stored image when loaded for execution.“ “These tests are sufficient to verify correct operation of cryptographic modules.”

177 For distributed TOEs the evaluator shall examine the TSS to ensure that it details which TOE component performs which self-tests and when these self-tests are run.

Findings:	[ST] / TOE Summary Specification states, “All TOE components (WLC and AP) run a suite of self-tests during initial start-up to verify correct operation of cryptographic modules.“ “During the system boot process (power on or reboot), all the Power on Startup Test (POST) components for all the cryptographic modules perform the POST for the corresponding component (hardware or software).” “All TOE components (WLC and AP) will automatically verify the integrity of the stored image when loaded for execution.“

2.5.3.2 Guidance Documentation

178 The evaluator shall also ensure that the guidance documentation describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.

Findings:	A description of the errors that may result from self-tests and the actions an administrator should take in response to those errors is found in the “Cryptographic Self-Tests” section of the [AGD]. Furthermore, The FPT_TST_EXT.1 section of Table 8 in the Auditing section of the [AGD] provides a description of the possible integrity errors that may occur when executing the self-test of the TOE. These descriptions correspond to the ones given in the FPT_TST_EXT.1 section of Table 21 in the TSS section of the [ST].

179 For distributed TOEs the evaluator shall ensure that the guidance documentation describes how to determine from an error message returned which TOE component has failed the self-test.

Findings:	The FPT_TST_EXT.1 section of Table 8 in the Auditing section of the [AGD] provides a description of the error message returned when a TOE component fails a self-test and how to identify that component from the error message. As per the description, the exact TOE component throwing the error can be identified using the “AP <AP name>” part of the error message.

2.5.3.3 Tests

180 It is expected that at least the following tests are performed:

a. Verification of the integrity of the firmware and executable software of the TOE

b. Verification of the correct operation of the cryptographic functions necessary to fulfil any of the SFRs.

181 Although formal compliance is not mandated, the self-tests performed should aim for a level of confidence comparable to:

a. [FIPS 140-2], chap. 4.9.1, Software/firmware integrity test for the verification of the integrity of the firmware and executable software. Note that the testing is not restricted to the cryptographic functions of the TOE.

b. [FIPS 140-2], chap. 4.9.1, Cryptographic algorithm test for the verification of the correct operation of cryptographic functions. Alternatively, national requirements of any CCRA member state for the security evaluation of cryptographic functions should be considered as appropriate.

182 The evaluator shall either verify that the self-tests described above are carried out during initial start-up or that the developer has justified any deviation from this.

High-Level Test Description
Reload the TOE and witness that the startup includes an indicator that self-tests were executed and passed permitting the device to operate. Manually run cryptographic self-tests and observe self-tests were run and were consistent with the description given in the TSS. Review vendor provided evidence and verify all self-tests are performed on initial start-up.
Findings: PASS

183 For distributed TOEs the evaluator shall perform testing of self-tests on all TOE components according to the description in the TSS about which self-test are performed by which component.

High-Level Test Description
Review vendor provided evidence and verify all self-tests are performed on initial start-up of TOE components (Access Points).
Findings: PASS

2.5.4 FPT_TUD_EXT.1 Trusted Update

2.5.4.1 TSS

184 The evaluator shall verify that the TSS describe how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the TSS needs to describe how and when the inactive version becomes active. The evaluator shall verify this description.

Findings:	[ST] / TOE Summary Specification states, “To query the currently active software version, the Administrator will need to navigate to the Dashboard page and locate the version listed under the Controller model in the top left corner. Alternatively, the same information can be obtained by entering the following command at the CLI: show version | include Cisco IOS XE Software” “For the APs, the Security Administrator can query the currently active AP software version by navigating to Monitoring -> Wireless -> AP Statistics. Clicking on an AP Name will display general AP information including the software version. Alternatively, the Administrator can enter the following command at the CLI: show ap image” “All images will not be active until the Administrator reboots the WLC as instructed in the CC Configuration Guide. When the WLC reboots the Access Points will automatically reboot.”

185 The evaluator shall verify that the TSS describes all TSF software update mechanisms for updating the system firmware and software (for simplicity the term 'software' will be used in the following although the requirements apply to firmware

and software). The evaluator shall verify that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails. Alternatively an approach using a published hash can be used. In this case the TSS shall detail this mechanism instead of the digital signature verification mechanism. The evaluator shall verify that the TSS describes the method by which the digital signature or published hash is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature or published hash of the update, and the actions that take place for both successful and unsuccessful signature verification or published hash verification.

Findings:	[ST] / TOE Summary Specification states, “Prior to being made publicly available, the software image is hashed using a SHA512 algorithm and then digitally signed. The digital signature is embedded to the image (hence the image is signed). The WLC uses a Cisco public key to validate the digital signature to obtain the SHA512 hash.” “The WLC then computes its own hash of the image using the same SHA512 algorithm. The WLC verifies the computed hash against the embedded hash. If they match the image has not been modified or tampered since distributed from Cisco meaning the software is authenticated. If they do not match the image will not install.” “AP software images are embedded in the WLC image and are not downloaded separately from Cisco.com.”

186 If the options ‘support automatic checking for updates’ or ‘support automatic updates’ are chosen from the selection in FPT_TUD_EXT.1.2, the evaluator shall verify that the TSS explains what actions are involved in automatic checking or automatic updating by the TOE, respectively.

Findings:	These options were not selected.

187 For distributed TOEs, the evaluator shall examine the TSS to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component. Alternatively, this description can be provided in the guidance documentation. In that case the evaluator should examine the guidance documentation instead.

Findings:	[ST] / TOE Summary Specification states, “AP software images are embedded in the WLC image and are not downloaded separately from Cisco.com. To keep software versions synchronized, after the WLC image has successfully transferred the CC Configuration Guide instructs the Administrator to download the AP image from the WLC image using a process termed AP preloading. The AP image is transferred over the DTLS protected internal channel and the AP will perform a digital signature verification check on the image it receives from the WLC.” “All images will not be active until the Administrator reboots the WLC as instructed in the CC Configuration Guide. When the WLC reboots the Access Points will automatically reboot.”

188 If a published hash is used to protect the trusted update mechanism, then the evaluator shall verify that the trusted update mechanism does involve an active authorization step of the Security Administrator, and that download of the published hash value, hash comparison and update is not a fully automated process involving no active authorization by the Security Administrator. In particular, authentication as Security Administration according to FMT_MOF.1/ManualUpdate needs to be part of the update process when using published hashes.

Findings:	The option to use published hash to protect the trusted update mechanism was not selected.

2.5.4.2 Guidance Documentation

189 The evaluator shall verify that the guidance documentation describes how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the guidance documentation needs to describe how to query the loaded but inactive version.

Findings:	The Verify TOE Software subsection of the section, Preparative Procedures and Operational Guidance for the TOE and the Update WLC and AP Software subsection of the section, Operational Guidance for the TOE in the [AGD] provide instructions on querying the active version of the TOE and TOE component software using the “show version”, “show install summary”, and “show ap image” commands. The “show ap image” command can also be used to query loaded but inactive versions of the TOE component software as well. Additionally, TOE software can be installed with a delayed activation as per the description given in the Update WLC and AP Software subsection. Installed versions become active when the “install activate” and “install commit” commands are run and the TOE reboots.

190 The evaluator shall verify that the guidance documentation describes how the verification of the authenticity of the update is performed (digital signature verification or verification of published hash). The description shall include the procedures for successful and unsuccessful verification. The description shall correspond to the description in the TSS.

Findings:	The Update WLC and AP Software section provides a description of on manual verification of the digital signatures of sub-packages. The section states, “If desired, the authorized administrator can manually verify the digital signature on each individual sub-package by executing verify bootflash:<package name> on the WLC. For example: WLC# verify bootflash: C9800-L-rpboot.17.03.02.SPA.pkg WLC# verify bootflash: C9800-L-mono-universalk9_wlc.17.03.02.SPA.pkg”

191 If a published hash is used to protect the trusted update mechanism, the evaluator shall verify that the guidance documentation describes how the Security Administrator can obtain authentic published hash values for the updates.

Findings:	A published hash is not used to protect the trusted update mechanism.

192 For distributed TOEs the evaluator shall verify that the guidance documentation describes how the versions of individual TOE components are determined for FPT_TUD_EXT.1, how all TOE components are updated, and the error conditions that may arise from checking or applying the update (e.g. failure of signature verification, or exceeding available storage space) along with appropriate recovery actions. The guidance documentation only has to describe the procedures relevant for the Security Administrator; it does not need to give information about the internal communication that takes place when applying updates.

Findings:	The Update WLC and AP Software subsection of the section, Operational Guidance for the TOE in the [AGD] provides instructions on querying the active version of the

	TOE component software “show ap image” as well as how to perform TOE component updates. TOE component software is bundled with the TOE software. As such, verification of TOE component software is done as a part of TOE software verification.

193 If this was information was not provided in the TSS: For distributed TOEs, the evaluator shall examine the Guidance Documentation to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component.

Findings:	This information is provided in the FPT_TUD_EXT.1 section of Table 21 of the TSS in the [ST].

194 If this was information was not provided in the TSS: If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the Guidance Documentation contains a description of how the certificates are contained on the device. The evaluator also ensures that the Guidance Documentation describes how the certificates are installed/updated/selected, if necessary.

Findings:	A certificate-based mechanism is not used for software update digital signature verification.

2.5.4.3 Tests

195 The evaluator shall perform the following tests:

a. Test 1: The evaluator performs the version verification activity to determine the current version of the product. If a trusted update can be installed on the TOE with a delayed activation, the evaluator shall also query the most recently installed version (for this test the TOE shall be in a state where these two versions match). The evaluator obtains a legitimate update using procedures described in the guidance documentation and verifies that it is successfully installed on the TOE. For some TOEs loading the update onto the TOE and activation of the update are separate steps (‘activation’ could be performed e.g. by a distinct activation step or by rebooting the device). In that case the evaluator verifies after loading the update onto the TOE but before activation of the update that the current version of the product did not change but the most recently installed version has changed to the new product version. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update and that current version of the product and most recently installed version match again.

High-Level Test Description
Get the current version of the TOE. Attempt to install a legitimate version of the TOE for the following circumstances: a downgrade, a “same-grade”, an upgrade. Stage the firmware in at least one case before activating it. After the install, get the current version of the TOE and ensure it is consistent with the newly installed version.
Findings: PASS

b. Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator first confirms that no updates are pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator obtains or produces illegitimate updates as defined below and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates:

    1) A modified version (e.g. using a hex editor) of a legitimately signed update

    2) An image that has not been signed

    3) An image signed with an invalid signature (e.g. by using a different key as expected for creating the signature or by manual modification of a legitimate signature)

    4) If the TOE allows a delayed activation of updates the TOE must be able to display both the currently executing version and most recently installed version. The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify, that both, current version and most recently installed version, reflect the same version information as prior to the update attempt.

High-Level Test Description
Attempt to install a bad image, an unsigned image and a badly signed image for both downgrades and upgrades. After each attempt, get the current version of the TOE using all available means and ensure they are consistent. Verify the images are rejected by the TOE and the installed/running firmware version information does not change.
Findings: PASS

c. Test 3 [conditional]: If the TOE itself verifies a hash value over an image against a published hash value (i.e. reference value) that has been imported to the TOE from outside such that the TOE itself authorizes the installation of an image to update the TOE, the following test shall be performed (otherwise the test shall be omitted. If the published hash is provided to the TOE by the Security Administrator and the verification of the hash value over the update file(s) against the published hash is performed by the TOE, then the evaluator shall perform the following tests. The evaluator first confirms that no update is pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test.

    1) The evaluator obtains or produces an illegitimate update such that the hash of the update does not match the published hash. The evaluator provides the published hash value to the TOE and

calculates the hash of the update either on the TOE itself (if that functionality is provided by the TOE), or else outside the TOE. The evaluator confirms that the hash values are different, and attempts to install the update on the TOE, verifying that this fails because of the difference in hash values (and that the failure is logged). Depending on the implementation of the TOE, the TOE might not allow the Security Administrator to even attempt updating the TOE after the verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE

2) The evaluator uses a legitimate update and tries to perform verification of the hash value without providing the published hash value to the TOE. The evaluator confirms that this attempt fails. The evaluator confirms that this attempt fails. Depending on the implementation of the TOE it might not be possible to attempt the verification of the hash value without providing a hash value to the TOE, e.g. if the hash value needs to be handed over to the TOE as a parameter in a command line message and the syntax check of the command prevents the execution of the command without providing a hash value. In that case the mechanism that prevents the execution of this check shall be tested accordingly, e.g. that the syntax check rejects the command without providing a hash value, and the rejection of the attempt is regarded as sufficient verification of the correct behaviour of the TOE in failing to verify the hash. The evaluator then attempts to install the update on the TOE (in spite of the unsuccessful hash verification) and confirms that this fails. Depending on the implementation of the TOE, the TOE might not allow to even attempt updating the TOE after the verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE

3) If the TOE allows delayed activation of updates, the TOE must be able to display both the currently executing version and most recently installed version. The handling of version information of the most recently installed version might differ between different TOEs. Depending on the point in time when the attempted update is rejected, the most recently installed version might or might not be updated. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify, that both, current version and most recently installed version, reflect the same version information as prior to the update attempt.

196 If the verification of the hash value over the update file(s) against the published hash is not performed by the TOE, Test 3 shall be skipped.

Findings:	The TOE does not support published hashes for image verification.

197 The evaluator shall perform Test 1, Test 2 and Test 3 (if applicable) for all methods supported (manual updates, automatic checking for updates, automatic updates).

Findings:	The TOE only supports manual updates. The test cases above are not applicable to automatic checking of updates since there are no images to install during an automatic check.

198 For distributed TOEs the evaluator shall perform Test 1, Test 2 and Test 3 (if applicable) for all TOE components.

Findings:	AP firmware images for all claimed models are embedded within the WLC firmware image and share the same version. In the evaluated configuration, the firmware of the APs can only be updated through the WLC. The WLC automatically updates AP firmware upon successful join of the AP to the WLC, ensuring the AP firmware version matches that of the WLC. Thus, Tests 1-3 for TOE components are satisfied by Tests 1-3 for the WLC (see above).

2.5.5 FPT_STM_EXT.1 Reliable Time Stamps

2.5.5.1 TSS

[Updated per TD 0632]

199 The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time, and that it provides a description of how the time is maintained and considered reliable in the context of each of the time related functions. If "obtain time from the underlying virtualization system" is selected, the evaluator shall examine the TSS to ensure that it identifies the VS interface the TOE uses to obtain time. If there is a delay between updates to the time on the VS and updating the time on the TOE, the TSS shall identify the maximum possible delay.

Findings:	[ST] / TOE Summary Specification states, "The TSF implements a clock function to provide a source of date and time. The clock function is reliant on the system clock provided by the underlying hardware. All controller models have a real-time clock (RTC) with battery to maintain time across reboots and power loss. APs synchronize their time with the WLC upon successfully joining." "The TOE relies upon date and time information for the following security functions: ■ To deny establishment of connections from wireless clients based on a configured time restriction (FTA_TSE.1); ■ To monitor local and remote interactive administrative sessions for inactivity (FTA_SSL_EXT.1, FTA_SSL.3); ■ Validating X.509 certificates to determine if a certificate has expired (FIA_X509_EXT.1/Rev, FIA_X509_EXT.1/ITT); ■ To determine when SSH session keys have expired and to initiate a rekey (FCS_SSHS_EXT.1); ■ To determine when IKEv2 SA lifetimes have expired and to initiate a rekey (FCS_IPSEC_EXT.1); ■ To determine when IKEv2 SA and Child SA lifetimes have expired and to initiate a rekey (FCS_IPSEC_EXT.1); ■ To provide accurate timestamps in audit records (FAU_GEN.1.2)."

2.5.5.2 Guidance Documentation

[Updated per TD 0632]

200 The evaluator examines the guidance documentation to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the guidance documentation instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication. If the TOE supports obtaining time from the underlying VS, the evaluator shall verify the Guidance Documentation specifies any configuration steps necessary. If no configuration is necessary, no statement is necessary in the Guidance Documentation. If there is a delay between updates to the

time on the VS and updating the time on the TOE, the evaluator shall ensure the Guidance Documentation informs the administrator of the maximum possible delay.

Findings:	The Configure Date and Time subsection of the section, Preparative Procedures and Operational Guidance for the TOE in the [AGD] provides instructions on setting the time. As described in the section, the time can be set on the TOE using the command, “clock set hh : mm : ss date month year”. The TOE does not leverage an NTP server in the evaluated configuration.

2.5.5.3 Tests

[Updated per TD 0632]

201 The evaluator shall perform the following tests:

a. Test 1: If the TOE supports direct setting of the time by the Security Administrator then the evaluator uses the guidance documentation to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.

High-Level Test Description
Using the CLI, change the date/time in the past by 1 day, 1 hour and 42 minutes. Verify the time was set properly. Using the CLI, change the date/time in the future by 7 days, 1 hour and 42 minutes. Verify the time was set properly. Repeat using the Web GUI.
Findings: PASS

b. Test 2: If the TOE supports the use of an NTP server; the evaluator shall use the guidance documentation to configure the NTP client on the TOE and set up a communication path with the NTP server. The evaluator will observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the guidance documentation.

Findings:	The TOE does not claim NTP.

c. Test 3: [conditional] If the TOE obtains time from the underlying VS, the evaluator shall record the time on the TOE, modify the time on the underlying VS, and verify the modified time is reflected by the TOE. If there is a delay between the setting the time on the VS and when the time is reflected on the TOE, the evaluator shall ensure this delay is consistent with the TSS and Guidance.

Findings:	The TOE does not obtain the time from the underlying VS.

202 If the audit component of the TOE consists of several parts with independent time information, then the evaluator shall verify that the time information between the different parts are either synchronized or that it is possible for all audit information to relate the time information of the different part to one base information unambiguously.

Findings:	The TOE does not support independent time information.

2.6 TOE Access (FTA)

2.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

2.6.1.1 TSS

203 The evaluator shall examine the TSS to determine that it details whether local administrative session locking or termination is supported and the related inactivity time period settings.

Findings:	[ST] / TOE Summary Specification states, “The Administrator can configure maximum inactivity times individually for both local and remote administrative sessions. If either the local or remote administrative sessions are inactive for a configured period of time, the session will be terminated and will require re-authentication.”

2.6.1.2 Guidance Documentation

204 The evaluator shall confirm that the guidance documentation states whether local administrative session locking or termination is supported and instructions for configuring the inactivity time period.

Findings:	The Session Termination subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] states, “All sessions at the local console and auxiliary port must terminate after an Administrator specified time interval of session inactivity has elapsed.” and provides instructions on how to configure the inactivity time period. As described in the section, this can be done using the command, “exec-timeout <time in minutes>” from within the line configuration sub-menu.

2.6.1.3 Tests

205 The evaluator shall perform the following test:

a. Test 1: The evaluator follows the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a local interactive session with the TOE. The evaluator then observes that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator then ensures that re-authentication is needed when trying to unlock the session.

High-Level Test Description
For each of 10, 12 minutes:
Change the idle timeout to this value;
Log into the device;

High-Level Test Description
With 30 seconds before the timeout expires, verify the session is still alive by sending a keep alive as described above in the TSFI commands. This should reset the timeout clock. The purpose is to ensure the timeout is not premature. Wait another minute. Verify the session is still alive by sending a keep alive. This should reset the timeout clock. The purpose is to ensure the timeout has been reset by the initial keep alive action above. Wait for the full duration of the timeout without sending any keep alives. The session should terminate.
Findings: PASS

2.6.2 FTA_SSL.3 TSF-initiated Termination

2.6.2.1 TSS

206 The evaluator shall examine the TSS to determine that it details the administrative remote session termination and the related inactivity time period.

Findings:	[ST] / TOE Summary Specification states, "The Administrator can configure maximum inactivity times individually for both local and remote administrative sessions. If either the local or remote administrative sessions are inactive for a configured period of time, the session will be terminated and will require re-authentication."

2.6.2.2 Guidance Documentation

207 The evaluator shall confirm that the guidance documentation includes instructions for configuring the inactivity time period for remote administrative session termination.

Findings:	The Remote Administration Protocols subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] provides instructions on how to configure the inactivity time period for SSH and HTTPS. For SSH, the subsection states, "16. Specify a timeout value for vty lines 0-4 WLC(config-line)# exec-timeout <time in minutes>". Where VTY lines 0-4 are configured for SSH in previous steps**.** For HTTPS, the subsection states, "All HTTPS sessions must terminate after an Administrator-configurable time interval of session inactivity has elapsed. Specify the timeout value in seconds. The range is from 180 to 1200. WLC(config)# ip http session-idle-timeout <180-1200>".

2.6.2.3 Tests

208 For each method of remote administration, the evaluator shall perform the following test:

a. Test 1: The evaluator follows the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a remote interactive session with the TOE. The evaluator then observes that the session is terminated after the configured time period.

High-Level Test Description
For each of 2, 3 minutes: Change the idle timeout to this value; Log into the device; With 30 seconds before the timeout expires, verify the session is still alive by sending a keep alive as described above in the TSFI commands. This should reset the timeout clock. The purpose is to ensure the timeout is not premature. Wait another minute. Verify the session is still alive by sending a keep alive. This should reset the timeout clock. The purpose is to ensure the timeout has been reset by the initial keep alive action above. Wait for the full duration of the timeout without sending any keep alives. The session should terminate. Note that the Web GUI timeout values tested were 3 and 4 minutes due to limitations on the minimum timeout value for the Web GUI.
Findings: PASS

2.6.3 FTA_SSL.4 User-initiated Termination

2.6.3.1 TSS

209 The evaluator shall examine the TSS to determine that it details how the local and remote administrative sessions are terminated.

Findings:	[ST] / TOE Summary Specification states, “The Administrator can terminate their own administrative sessions. The Administrator can logout of the Web GUI by clicking logout icon in the top-right corner of the page. The Administrator can logout of the CLI by entering the `logout` or `exit` command.”

2.6.3.2 Guidance Documentation

210 The evaluator shall confirm that the guidance documentation states how to terminate a local or remote interactive session.

Findings:	The Access Remote Administrative Interfaces subsection of the section, Operational Guidance for the TOE of the [AGD] provides instructions on how to terminate remote and local CLI interactive sessions as well as remote Web GUI sessions. For SSH, the subsection states, “logout out of your local console CLI session by entering either “exit or “logout” WLC# logout”. For HTTPS, the Access Remote Administrative Interfaces subsection of the section, Operational Guidance for the TOE of the [AGD] states:

	"To logout click the exit icon in the top right corner:"

2.6.3.3 Tests

211 For each method of remote administration, the evaluator shall perform the following tests:

a. Test 1: The evaluator initiates an interactive local session with the TOE. The evaluator then follows the guidance documentation to exit or log off the session and observes that the session has been terminated.

High-Level Test Description
Log into the serial console Log out using the TSFI previous discussed. Verify that the session has been terminated.
Findings: PASS

b. Test 2: The evaluator initiates an interactive remote session with the TOE. The evaluator then follows the guidance documentation to exit or log off the session and observes that the session has been terminated.

High-Level Test Description
Log into the SSH CLI interface and log out. Verify the session is terminated. Log into the Web GUI interface and copy the URL presented. Log out using the TSFI previous discussed. Paste the URL back into the web browser and attempt to navigate directly to it. Verify the session is terminated.
Findings: PASS

2.6.4 FTA_TAB.1 Default TOE Access Banners

2.6.4.1 TSS

212 The evaluator shall check the TSS to ensure that it details each administrative method of access (local and remote) available to the Security Administrator (e.g., serial port, SSH, HTTPS). The evaluator shall check the TSS to ensure that all administrative methods of access available to the Security Administrator are listed and that the TSS states that the TOE is displaying an advisory notice and a consent warning message for each administrative method of access. The advisory notice and the consent warning message might be different for different administrative methods of access, and might be configured during initial configuration (e.g. via configuration file).

Findings:	[ST] / TOE Summary Specification states, "The Administrator can configure an access banner that describes restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the Controller. The banner will display on the local console port, SSH, and HTTPS interfaces prior to allowing any administrative access."

2.6.4.2 Guidance Documentation

213 The evaluator shall check the guidance documentation to ensure that it describes how to configure the banner message.

Findings:	The Access Banner subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] provides instructions on how to configure a banner message. As described in the subsection, the command, “banner login z <message text> z” where “z” is the chosen delimiter character can be used for this purpose.

2.6.4.3 Tests

214 The evaluator shall also perform the following test:

a. Test 1: The evaluator follows the guidance documentation to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

High-Level Test Description
Log into the CLI interface. Change the banner to a random string. Log into fresh sessions for all interactive interfaces and show that the banner was modified and is presented prior to I&A.
Findings: PASS

2.7 Trusted path/channels (FTP)

2.7.1 FTP_ITC.1 Inter-TSF trusted channel

2.7.1.1 TSS

215 The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each secure communication mechanism is identified in terms of the allowed protocols for that IT entity, whether the TOE acts as a server or a client, and the method of assured identification of the non-TSF endpoint. The evaluator shall also confirm that all secure communication mechanisms are described in sufficient detail to allow the evaluator to match them to the cryptographic protocol Security Functional Requirements listed in the ST.

Findings:	[ST] / TOE Summary Specification includes the following table:

TOE Component	Acting as Client or Server	IT Entity	Secure Communication Mechanism/ Protocol	Non-TSF Endpoint Identification
WLC	Client	Syslog Server	IPsec	X.509 Certificate
WLC	Client	RADIUS Server	RADsec	X.509 Certificate
WLC	Client	RADIUS Server	IEEE 802.1X	X.509 Certificate
WLC	Client	EST Server	TLS	X.509 Certificate
AP	Server	Wireless Client	IEEE 802.11-2012 (WPA2)	IEEE 802.1X EAP-TLS

2.7.1.2 Guidance Documentation

216 The evaluator shall confirm that the guidance documentation contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken.

Findings: The TOE uses IPsec to communicate with an external Syslog server and TLS to communicate with external RADIUS and EST servers.

Instructions on how to configure these connections are present within the associated section of the [AGD], namely, the IPsec, TLS-RADsec and CC Mode subsections of the section, Preparative Procedures and Operational Guidance for the TOE. Further EST server configuration instructions are found in [EST_REF].

Furthermore, the IPsec Session Interruption and Recovery subsection of the section, Operational Guidance for the TOE, of the [AGD] states, “When a connection is broken no administrative interaction is required. The IPsec session will be re-established (a new SA set up) once the peer is back online.”

RADSec Session Interruption and Recovery subsection of the Operational Guidance for the TOE section, of the [AGD] states, “If a RADsec connection is unexpectedly interrupted, the TLS client connection will be broken and the Administrator will find a the state listed as DOWN in the output of show aaa servers command.

“If this condition occurs no administrative interaction is required. The RADsec session will be reestablished and a new TLS client session setup once the peer is back online.”

EST Server Session Interruption and Recovery subsection of the Operational Guidance for the TOE section, of the [AGD] states, “If an EST Server connection is unexpectedly interrupted during certificate enrollment, the TLS client connection will be broken and the Administrator will find the LSC provisioning for Access Points has failed. Specifically, the Access Point will not automatically reboot.

	“If this condition occurs the administrator will need to re-perform steps 6 - 10 in the "Enable LSC Provisioning for AP" section of this document once the EST server peer is back online.”

2.7.1.3 Tests

217 The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report.

218 The evaluator shall perform the following tests:

a. Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

Findings:	The TOE maintains trusted channels to the RADIUS and EST servers via TLS and to the Syslog server via IPSec, which are set up as per the evaluated configuration. These channels are tested throughout the evaluation. It should also be noted that secure communications with the EST and RADIUS servers are setup as needed and torn down immediately after use. Test 1, 2 and 3, outlined here, are performed for each secure communication channel in the following sections: - EST Server (TLS) - FCS_TLSC_EXT.1 (EST) - Radius Server (TLS) - FCS_TLSC_EXT.1 (RadSec) - Syslog Server (IPSec) – FCS_IPSEC_EXT.1

b. Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be initiated from the TOE.

Findings:	See previous test case.

c. Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.

Findings:	See previous test case.

d. Test 4: Objective: The objective of this test is to ensure that the TOE reacts appropriately to any connection outage or interruption of the route to the external IT entities.

The evaluator shall, for each instance where the TOE acts as a client utilizing a secure communication mechanism with a distinct IT entity, physically interrupt the connection of that IT entity for the following durations: i) a duration that exceeds the TOE’s application layer timeout setting, ii) a

duration shorter than the application layer timeout but of sufficient length to interrupt the network link layer.

The evaluator shall ensure that, when the physical connectivity is restored, communications are appropriately protected and no TSF data is sent in plaintext.

In the case where the TOE is able to detect when the cable is removed from the device, another physical network device (e.g. a core switch) shall be used to interrupt the connection between the TOE and the distinct IT entity. The interruption shall not be performed at the virtual node (e.g. virtual switch) and must be physical in nature.

High-Level Test Description
Log into the TOE and engage wireshark to capture traffic on the appropriate interface. Initiate a secure connection from the TOE to the appropriate IT entity using the commands specified above. Physically disconnect the TOE from the environment and immediately reconnect (or wait a duration shorter than the TOE's application layer timeout setting before reconnecting). Examine wireshark to verify that the protected communications have not been affected. Physically disconnect the TOE from the environment and wait for 30 minutes (or a duration longer than the TOE's application layer timeout setting) and then physically reconnect the TOE back to the network. Examine wireshark to verify that the protected interface has automatically re-established encrypted communications without any user intervention. Note that for EST and RADIUS communications, the administrator must manually trigger connection re-establishment as such secure channels are ephemeral in nature and are typically established on a per-transaction basis.
Findings: PASS

Further assurance activities are associated with the specific protocols.

219 For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of external secure channels to TOE components in the Security Target.

High-Level Test Description
Engage wireshark to capture traffic on the appropriate interface. Initiate a secure connection between the TOE component (AP) and TOE (WLC). Physically disconnect the TOE from the environment and immediately reconnect. Examine wireshark to verify that the protected communications have not been affected. Physically disconnect the TOE from the environment and wait for 30 minutes (or a duration longer than the TOE's application layer timeout setting) and then physically reconnect the TOE back to the network. Note for DTLS communications between TOE components, timeout typically occurs after 30 seconds. Examine wireshark to verify that the protected interface has automatically re-established encrypted communications without any user intervention.
Findings: PASS

220 The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public- facing document or report.

2.7.2 FTP_TRP.1/Admin Trusted Path

2.7.2.1 TSS

221 The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

Findings:	[ST] / TOE Summary Specification states, "All remote administrative communications take place over a secure encrypted SSHv2 (CLI) session or HTTPS (web-based GUI) session. Both SSHv2 and HTTPS sessions are protected using AES encryption." The evaluator confirmed that all protocols listed in the TSS are consistent with the requirement and the requirements in the ST.

2.7.2.2 Guidance Documentation

222 The evaluator shall confirm that the guidance documentation contains instructions for establishing the remote administrative sessions for each supported method.

Findings:	Instructions for establishing remote administrative sessions with the TOE over SSH and HTTPS are provided in the Access Remote Administrative Interfaces subsection of the section, Operational Guidance for the TOE of the [AGD]. This section states, "From your remote management workstation, initiate a connect using SSH and supply either your public key or password credentials. Upon successful login you will be presented with privilege administrator access denoted by the 'hashtag' symbol:" and "From the Management workstation open a web browser to the IP address or fully-qualified domain name of the Controller. To login use the username and password credentials as for CLI/SSH."

2.7.2.3 Tests

223 The evaluator shall perform the following tests:

a. Test 1: The evaluators shall ensure that communications using each specified (in the guidance documentation) remote administration method is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

Findings:	The only trusted paths are the Web GUI and SSH CLI, which are both set up as per the evaluated configuration. They are constantly tested throughout the evaluation.

b. Test 2: The evaluator shall ensure, for each communication channel, the channel data is not sent in plaintext.

Findings:	This test is performed in conjunction with FCS_TLSS_EXT and FCS_SSHS_EXT testing.

224 Further assurance activities are associated with the specific protocols.

225 For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of trusted paths to TOE components in the Security Target.

Findings:	This test is performed in conjunction with FCS_DTLSS_EXT.1 and FCS_DTLSC_EXT.1 testing.

3 Evaluation Activities for NDcPP Optional Requirements

3.1 Security Audit (FAU)

3.1.1 FAU_STG.1 Protected audit trail storage

3.1.1.1 TSS

226 The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally and how these records are protected against unauthorized modification or deletion. The evaluator shall ensure that the TSS describes the conditions that must be met for authorized deletion of audit records.

Findings:	[ST] / TOE Summary Specification states, ", the TOE will buffer between 4096-bytes and 2,148,483,647 bytes of audit records on the TOE." "The WLC protects the local logging buffer from unauthorized access, modification or deletion. No account is able to modify data that has been written to the local logging buffer. Only the Administrator is able to clear the local logging buffer."

227 For distributed TOEs the evaluator shall examine the TSS to ensure it describes to which TOE components this SFR applies and how local storage is implemented among the different TOE components (e.g. every TOE component does its own local storage or the data is sent to another TOE component for central local storage of all audit events).

Findings:	[ST] / TOE Summary Specification states, "The Cisco Catalyst 9800 Series Wireless Controllers and Access Points 17.6 TOE is distributed. After the AP joins the WLC to form a distributed TOE the AP will transmit its audit messages to the WLC over the secure DTLS channel described in FPT_ITT.1."

3.1.1.2 Guidance Documentation

228 The evaluator shall examine the guidance documentation to determine that it describes any configuration required for protection of the locally stored audit data against unauthorized modification or deletion.

Findings:	The [AGD] does not identify any configuration necessary to protect the locally stored audit data from unauthorized modification or deletion. No interface is provided by the TOE to modify such data.

3.1.1.3 Tests

The evaluator shall perform the following tests:

229 a) Test 1: The evaluator shall access the audit trail without authentication as Security Administrator (either by authentication as a nonadministrative user, if supported, or without authentication at all) and attempt to modify and delete the audit records. The evaluator shall verify that these attempts fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to access the audit trail can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

High-Level Test Description
Login to the TOE as a privileged user and view the locally stored audit records. Logout of the TOE and log back in as a non-privileged user. Attempt to delete locally stored audit logs using the above command. Logout of the TOE and log back in as the Security Administrator. View the logs using the above command and confirm they are unchanged (with the exception of any additional login/logout events for the non-privileged user). Repeat using the Web GUI.
Findings: PASS

230 b) Test 2: The evaluator shall access the audit trail as an authorized administrator and attempt to delete the audit records. The evaluator shall verify that these attempts succeed. The evaluator shall verify that only the records authorized for deletion are deleted.

High-Level Test Description
Login to the TOE as a privileged user and view the locally stored audit records. Attempt to delete locally stored audit logs using the above command. View the logs using the above command and confirm they have been deleted. Repeat using the Web GUI.
Findings: PASS

231 For distributed TOEs the evaluator shall perform test 1 and test 2 for each component that is defined by the TSS to be covered by this SFR.

Findings:	The TSS does not describe any components other than the WLC that store audit records locally.

3.2 Communication (FCO)

3.2.1 FCO_CPC_EXT.1 Component Registration Channel Definition

3.2.1.1 TSS

232 (Note: paragraph 274 (of the [SD]) lists questions for which the evaluator needs to determine and report answers through the combination of the TSS, Guidance Documentation, and Tests Evaluation Activities.)

233 The evaluator shall examine the TSS to confirm that it:

a) Describes the method by which a Security Administrator enables and disables communications between pairs of TOE components.

b) Describes the relevant details according to the type of channel in the main selection made in FCO_CPC_EXT.1.2:

• First type: the TSS identifies the relevant SFR iteration that specifies the channel used

• Second type: the TSS (with support from the operational guidance if selected in FTP_TRP.1.3/Join) describes details of the channel and the mechanisms that it uses (and describes how the process ensures that the key is unique to the pair of components) – see also the Evaluation Activities for FTP_TRP.1/Join.

Findings:	[ST] / TOE Summary Specification describes the following: a) "At the WLC, before an AP can join and communicate with a WLC, the Administrator must enable an AP authorization list maintained on the WLC. The AP authorization list defines the APs that are permitted to join by identification of its unique serial number. The AP authorization list is available under Configuration -> Security -> AAA Advanced in the Controller GUI. Only the Administrator can access the AP authorization list." b) First type: "All aspects of the registration and internal communication channel are met by FPT_ITT.1. Refer to FCS_DTLSS_EXT.1 for additional information."

234 The evaluator shall confirm that if any aspects of the registration channel are identified as not meeting FTP_ITC.1 or FPT_ITT.1, then the ST has also selected the FTP_TRP.1/Join option in the main selection in FCO_CPC_EXT.1.2.

Findings:	The evaluator confirmed that all aspects meet FPT_ITT.1.

3.2.1.2 Guidance Documentation

235 (Note: paragraph 274 (of the [SD]) lists questions for which the evaluator needs to determine and report answers through the combination of the TSS, Guidance Documentation, and Tests Evaluation Activities.)

236 The evaluator shall examine the guidance documentation to confirm that it contains instructions for enabling and disabling communications with any individual component of a distributed TOE. The evaluator shall confirm that the method of disabling is such that all other components can be prevented from communicating with the component that is being removed from the TOE (preventing the remaining components from either attempting to initiate communications to the disabled component, or from responding to communications from the disabled component).

Findings:	The Adding New APs and Enable/Disable APs subsections of the section, Operational Guidance for the TOE and the DTLS-CAPWAP subsection of the section, Preparative Procedures and Operational Guidance for the TOE in the [AGD] provide instructions for enabling/disabling communications with TOE components. The Enable/Disable APs subsection of the section, Operational Guidance for the TOE states, "At any point the Administrator may enable or disable APs from joining. In the Web GUI, Navigate to Configuration -> Security -> AAA. Click on AAA Advanced - Device Authentication. Click the Serial Number tab and add the AP Serial Number. To remove an AP click the checkbox next to the AP and then click the Delete button." TOE components do not communicate directly with one another.

237 The evaluator shall examine the guidance documentation to confirm that it includes recovery instructions should a connection be unintentionally broken during the registration process.

Findings:	As per the sections of the [AGD] listed above, TOE component registration occurs automatically through the TOE using the configured method.

	Furthermore, the DTLS Session Interruption and Recovery subsection of the section, Operational Guidance for the TOE of the [AGD] states, "If this condition occurs the AP will restart the DTLS connection and attempt to re-join the WLC automatically. No Security Administrator intervention is required for the AP to recover from an interrupted DTLS session."

238 If the TOE uses a registration channel for registering components to the TOE (i.e. where the ST author uses the FTP_ITC.1/FPT_ITT.1 or FTP_TRP.1/Join channel types in the main selection for FCO_CPC_EXT.1.2) then the evaluator shall examine the Preparative Procedures to confirm that they:

a) describe the security characteristics of the registration channel (e.g. the protocol, keys and authentication data on which it is based) and shall highlight any aspects which do not meet the requirements for a steady-state inter-component channel (as in FTP_ITC.1 or FPT_ITT.1)

b) identify any dependencies between the configuration of the registration channel and the security of the subsequent inter-component communications (e.g. where AES-256 inter-component communications depend on transmitting 256 bit keys between components and therefore rely on the registration channel being configured to use an equivalent key length)

c) identify any aspects of the channel can be modified by the operational environment in order to improve the channel security and shall describe how this modification can be achieved (e.g. generating a new key pair, or replacing a default public key certificate).

Findings:	The evaluator reviewed the preparative procedures in the [AGD] and confirmed that the security characteristics of the registration channel are described, discrepancies between the configuration of the registration channel and the security of the subsequent inter-component communications are identified, and aspects of the channel that can be modified by the operational environment in order to improve channel security and how to do so are identified. Furthermore, The DTLS – CAPWAP subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] states, "The first time an Access Point joins the Controller it must use either a manufactured-installed certificate (MIC) or a self-signed certificate (SSC). MICs and SSCs are only for the very first time the AP joins a Controller. For all subsequent joins, the AP will use Locally Significant Certificates (LSC)."

239 As background for the examination of the registration channel description, it is noted that the requirements above are intended to ensure that administrators can make an accurate judgement of any risks that arise from the default registration process. Examples would be the use of self-signed certificates (i.e. certificates that are not chained to an external or local Certification Authority), manufacturer-issued certificates (where control over aspects such as revocation, or which devices are issued with recognised certificates, is outside the control of the operational environment), use of generic/non-unique keys (e.g. where the same key is present on more than one instance of a device), or well-known keys (i.e. where the confidentiality of the keys is not intended to be strongly protected – note that this need not mean there is a positive action or intention to publicise the keys).

Findings:	The DTLS-CAPWAP, FIPS Mode and CC Mode subsections of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] describe how component registration is accomplished (see above quoted section). The initial registration channel relies on either self-signed certificates (SSC) or manufacturer-

issued certificates (MIC) as well as authorized TOE component serial numbers configured on the TOE.

The DTLS – CAPWAP subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] states,

“The first time an Access Point joins the Controller it must use either a manufactured-installed certificate (MIC) or a self-signed certificate (SSC). MICs and SSCs are only for the very first time the AP joins a Controller. For all subsequent joins, the AP will use Locally Significant Certificates (LSC).”

The FIPS Mode subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] describes the usage of the “ap auth-list authorize-serialNum” and “username <AP serial number> serial-number commands to authorize APs by serial number.

Based on the description given in the above subsections of the [AGD], the evaluator determined an administrator can make an accurate judgement of any risks that arise from the default registration process.

240 In the case of a distributed TOE for which the ST author uses the FTP_TRP.1/Join channel type in the main selection for FCO_CPC_EXT.1.2 and the TOE relies on the operational environment to provide security for some aspects of the registration channel security then there are additional requirements on the Preparative Procedures as described in section 3.4.1.2.

Findings:	This selection is not made in the [ST].

3.2.1.3 Tests

241 (Note: paragraph 274 (of the [SD]) lists questions for which the evaluator needs to determine and report answers through the combination of the TSS, Guidance Documentation, and Tests Evaluation Activities.)

242 The evaluator shall carry out the following tests:

243 a) Test 1.1: the evaluator shall confirm that an IT entity that is not currently a member of the distributed TOE cannot communicate with any component of the TOE until the non-member entity is enabled by a Security Administrator for each of the non-equivalent TOE components[3] that it is required to communicate with (non-equivalent TOE components are as defined in the minimum configuration for the distributed TOE)

High-Level Test Description
Add an unauthorized AP to the WLAN network of the WLC. Verify the AP is unable to join the controller. Attempt to run a remote command on the unauthorized AP.Verify the attempt fails. Authorize the AP to join the WLC. Verify the AP is able to join the controller.

[3] An ‘equivalent TOE component’ is a type of distributed TOE component that exhibits the same security characteristics, behaviour and role in the TSF as some other TOE component. In principle a distributed TOE could operate with only one instance of each equivalent TOE component, although the minimum configuration of the distributed TOE may include more than one instance (see discussion of the minimum configuration of a distributed TOE, in section A.9). In practice a deployment of the TOE may include more than one instance of some equivalent TOE components for practical reasons, such as performance or the need to have separate instances for separate subnets or VLANs.

High-Level Test Description
Attempt to run a remote command on the authorized AP.Verify the attempt succeeds. Optional: Using the Lightship DTLS client, attempt to join the controller with a valid device certificate with a CN that does not match an authorized AP serial-number. Verify the DTLS connection fails. Authorize an AP serial-number that corresponds to the device certificate. Verify the DTLS connection succeeds.
Findings: PASS

b) Test 1.2: the evaluator shall confirm that after enablement, an IT entity can communicate only with the components that it has been enabled for. This includes testing that the enabled communication is successful for the enabled component pair, and that communication remains unsuccessful with any other component for which communication has not been explicitly enabled

Some TOEs may set up the registration channel before the enablement step is carried out, but in such a case the channel must not allow communications until after the enablement step has been completed.

Findings:	This test was performed in Test 1.1.

244 The evaluator shall repeat Tests 1.1 and 1.2 for each different type of enablement process that can be used in the TOE.

c) Test 2: The evaluator shall separately disable each TOE component in turn and ensure that the other TOE components cannot then communicate with the disabled component, whether by attempting to initiate communications with the disabled component or by responding to communication attempts from the disabled component.

Findings:	This test was performed in Test 1.1.

d)Test 3: The evaluator shall carry out the following tests according to those that apply to the values of the main (outer) selection made in the ST for FCO_CPC_EXT.1.2.

1) If the ST uses the first type of communication channel in the selection in FCO_CPC_EXT.1.2 then the evaluator tests the channel via the Evaluation Activities for FTP_ITC.1 or FPT_ITT.1 according to the second selection – the evaluator shall ensure that the test coverage for these SFRs includes their use in the registration process.

2) If the ST uses the second type of communication channel in the selection in FCO_CPC_EXT.1.2 then the evaluator tests the channel via the Evaluation Activities for FTP_TRP.1/Join.

3) If the ST uses the ‘no channel’ selection, then no test is required.

Findings:	This is done as required.

e) Test 4: The evaluator shall perform one of the following tests, according to the TOE characteristics identified in its TSS and operational guidance:

1) If the registration channel is not subsequently used for inter-component communication, and in all cases where the second selection in FCO_CPC_EXT.1.2

is made (i.e. using FTP_TRP.1/Join) then the evaluator shall confirm that the registration channel can no longer be used after the registration process has completed, by attempting to use the channel to communicate with each of the endpoints after registration has completed

Test Not Applicable:	The registration channel is subsequently used for intercomponent communication. Additionally, the second selection is not made in FCO_CPC_EXT.1.2 of the ST.

2) If the registration channel is subsequently used for inter-component communication then the evaluator shall confirm that any aspects identified in the operational guidance as necessary to meet the requirements for a steady-state inter-component channel (as in FTP_ITC.1 or FPT_ITT.1) can indeed be carried out (e.g. there might be a requirement to replace the default key pair and/or public key certificate).

Findings:	The operational guidance does not identify any additionally requirements for a steady-state intercomponent channel beyond what is tested in FCS_DTLSS_EXT.1 and FIA_X509_EXT.1/ITT.

f) Test 5: For each aspect of the security of the registration channel that operational guidance states can be modified by the operational environment in order to improve the channel security (cf. AGD_PRE.1 refinement item 2 in (cf. the requirements on Preparative Procedures in 3.5.1.2), the evaluator shall confirm, by following the procedure described in the operational guidance, that this modification can be successfully carried out.

Findings:	The operational guidance does not identify any security aspects of the registration channel that can be made beyond what is tested in FCS_DTLSS_EXT.1, FCS_DTLSC_EXT.1, and FIA_X509_EXT.1/ITT.

3.3 Cryptographic Support (FCS)

3.3.1 FCS_TLSC_EXT.2 Extended: TLS Client support for mutual authentication

3.3.1.1 TSS

FCS_TLSC_EXT.2.1

245 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.

Findings:	[ST] / TOE Summary Specification states, “The TOE supports TLS mutual authentication and will present a client certificate to the RADsec server and EST Server during connection establishment.”

3.3.1.2 Guidance Documentation

FCS_TLSC_EXT.2.1

246 If the TSS indicates that mutual authentication using X.509v3 certificates is used, the evaluator shall verify that the AGD guidance includes instructions for configuring the client-side certificates for TLS mutual authentication.

Findings:	Instructions on configuring client-side certificates for TLS mutual authentication are provided in subsections of the [AGD] which require TLS client authentication, namely, TLS-RADsec and DTLS-CAPWAP/CC Mode/Enable LSC Provisioning for AP. The TLS-RADsec subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] states, "TLS must be configured to use X.509v3 certificates supporting a minimum path length of three (root CA -> intermediate CA -> end-entity). Therefore, you will need to create two trustpoints. The section below provides steps to create a root CA and a subordinate CA using CA certificates from your organization's PKI. Before proceeding, please have the root CA and subordinate CA certificates ready for import from your CA administrator. Note: The TOE may be configured to perform identity verification using either an IP address or DNS Name in the SAN extension of the X.509 certificate. This is covered in step 14 in the section below. The Administrator is advised to follow the security policies and procedures of their organization if using an IP address to verify RADsec server identity." The CC Mode subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] describes the configuration of "mysubESTCA" and "mysubESTCA-ecc" trustpoints for mutual authentication to the EST server.

3.3.1.3 Tests

247 For all tests in this chapter the TLS server used for testing of the TOE shall be configured to require mutual authentication.

[Updated per TD 0670]

FCS_TLSC_EXT.2.1

248 Test 1: The evaluator shall establish a connection to a peer server that is configured for mutual authentication (i.e. sends a server Certificate Request (type 13) message). The evaluator observes that the TOE TLS client sends both client Certificate (type 11) and client Certificate Verify (type 15) messages during its negotiation of a TLS channel and that Application Data is sent.

High-Level Test Description
Initiate a TLS connection from the TOE to the peer TLS server configured for mutual authentication and verify that the TOE sends an appropriate Certificate and Certificate Verify message in response to a Certificate Request sent by the server.
Findings: PASS

249 In addition, all other testing in FCS_TLSC_EXT.1 and FIA_X509_EXT.* must be performed as per the requirements.

3.4 Identification and Authentication (FIA)

3.4.1 FIA_X509_EXT.1/ITT X.509 Certificate Validation

3.4.1.1 TSS

250 The evaluator shall examine the TSS to ensure it describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied). If selected, the TSS shall describe how certificate revocation checking is performed. It is not sufficient to verify the status of a X.509 certificate only when it's loaded onto the device.

Findings:	[ST] / TOE Summary Specification states, "X.509v3 certificate validation is performed when the AP attempts to join the WLC. The AP will only be able to join the WLC and form a distributed TOE if the WLC determines the X.509v3 certificate of the AP is valid and the subject Distinguished Name field, which contains the AP's hardware serial number, matches an entry in the AP authorization list defined and maintained by the Security Administrator. The WLC will also verify the extendedKeyUsage field of the AP certificate contains the Client Authentication purpose." The evaluator identified that certificate revocation was not selected.

3.4.1.2 Guidance Documentation

251 The evaluator shall also ensure that the guidance documentation describes where the check of validity of the certificates takes place, describes any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) and describe how certificate revocation checking is performed.

Findings:	A description of where validity checking of certificates takes place is provided in the DTLS-CAPWAP, and CC Mode subsections of the section, Preparative Procedures and Operational Guidance for the TOE in the [AGD]. The [AGD] provides the following where the check of validity takes place for distributed TOE components in section "Enable LSC Provisioning for AP": "Note: The TOE uses X.509v3 certificates to support authentication for DTLS connections. X.509v3 certificate validation is performed when the AP attempts to join the WLC. The AP will only be able to join the WLC and form a distributed TOE if the WLC determines the X.509v3 certificate of the AP is valid and the subject Distinguished Name field, which contains the AP's hardware serial number, matches an entry in the AP authorization list defined and maintained by the Security Administrator. The WLC will also verify the extendedKeyUsage field of the AP certificate contains the Client Authentication purpose. OCSP is not supported; therefore the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) is trivially satisfied by the TOE." Revocation checking is not selected for ITT communications in the [ST].

3.4.1.3 Tests

252 The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the

TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.:

a) Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store).

Findings:	This test is covered by FCS_DTLSS_EXT.1.

Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.

High-Level Test Description
Remove the DTLS trust anchor from the WLC's trust store. Initiate a DTLS connection to the WLC using the Lightship Security DTLS test tool. Verify the WLC's attempt to validate the DTLS client certificate fails by inspection of the WLC's audit logs and associated traffic capture.
Findings: PASS

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

High-Level Test Description
Initiate a DTLS connection to the WLC using the Lightship Security DTLS test tool with an expired client certificate. Verify the WLC's attempt to validate the DTLS client certificate fails by inspection of the WLC's audit logs and associated traffic capture. Replace the DTLS trust anchor with a soon-to-expire, yet otherwise valid, CA certificate. Initiate a DTLS connection to the WLC using the Lightship Security DTLS test tool with a valid client certificate. Verify the WLC's attempt to validate the DTLS client certificate succeeds, prior to CA certificate expiry, by inspection of the WLC's audit logs and associated traffic capture. Initiate a DTLS connection to the WLC using the Lightship Security DTLS test tool with a valid client certificate. Verify the WLC's attempt to validate the DTLS client certificate fails, after the CA certificate is expires, by inspection of the WLC's audit logs and associated traffic capture.
Findings: PASS

c) Test 3: The evaluator shall test that the TOE can properly handle revoked certificates-–conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the TOE certificate and revocation of the TOE intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails.

No testing is required if no revocation method is selected. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.

Test Not Applicable:	The TOE does not claim support for certificate revocation checking for FIA_X509_EXT.1/ITT.

d) Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

Test Not Applicable:	The TOE does not claim support for certificate revocation checking for FIA_X509_EXT.1/ITT.

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

High-Level Test Description
Using the Lightship Security DTLS test tool, initiate a DTLS connection to the WLC using a client certificate containing the described modifications. Verify the WLC fails to validate the certificate by inspection of the WLC's audit logs and associated traffic capture.
Findings: PASS

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

High-Level Test Description
Using the Lightship Security DTLS test tool, initiate a DTLS connection to the WLC using a client certificate containing the described modifications. Verify the WLC fails to validate the certificate by inspection of the WLC's audit logs and associated traffic capture.
Findings: PASS

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

High-Level Test Description
Using the Lightship Security DTLS test tool, initiate a DTLS connection to the WLC using a client certificate containing the described modifications. Verify the WLC fails to validate the certificate by inspection of the WLC's audit logs and associated traffic capture.
Findings: PASS

Technical Decision: This test was added per TD0527.

253 h) Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The evaluator shall conduct the following tests:

Test 8a: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates (terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

Test Not Applicable: The TOE does not claim the ability to process CA certificates presented in certificate messages.

Test 8b: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

Test Not Applicable: The TOE does not claim the ability to process CA certificates presented in certificate messages.

Test 8c: The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store.

Findings: The trust store on the WLC is common to all trusted channels. Individual trustpoints must be configured for use with a given channel i.e. (EST, DTLS, IPsec etc.). The process of uploading a CA certificate to a trustpoint, and its subsequent validation, is common to all trusted channels. The generic version of this test is done in FIA_X509_EXT.1/Rev (EST).

254 The evaluator shall perform the following tests for FIA_X509_EXT.1.2/ITT. The tests described must be performed in conjunction with the other certificate services

assurance activities, including the functions in FIA_X509_EXT.2.1/ITT. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted.

255 The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation).

256 For each of the following tests the evaluator shall create a chain of at least two certificates: a self-signed root CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain).

a) Test 1: The evaluator shall ensure that one CA in the chain does not contain the basicConstraints extension. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

Findings:	The trust store on the WLC is common to all trusted channels. Individual trustpoints must be configured for use with a given channel i.e. (EST, DTLS, IPsec etc.). The process of uploading a CA certificate to a trustpoint, and its subsequent validation, is common to all trusted channels. The generic version of this test is done in FIA_X509_EXT.1/Rev (RadSec) in which the described certificate was shown to be rejected on upload to the TOE's trust store.

b) Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

Findings:	The trust store on the WLC is common to all trusted channels. Individual trustpoints must be configured for use with a given channel i.e. (EST, DTLS, IPsec etc.). The process of uploading a CA certificate to a trustpoint, and its subsequent validation, is common to all trusted channels. The generic version of this test is done in FIA_X509_EXT.1/Rev (RadSec) in which the described certificate was shown to be rejected on upload to the TOE's trust store.

3.5 Protection of the TSF (FPT)

3.5.1 FPT_ITT.1 Basic internal TSF data transfer protection

3.5.1.1 TSS

257 The evaluator shall examine the TSS to determine that, for all communications between components of a distributed TOE, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS for these inter-component communications are specified and included in the requirements in the ST.

Findings:	[ST] /TOE Summary Specification states, “The TOE includes two distinct types of components that use a secure network protocol for internal communication. When TSF data is transferred between APs and WLCs the data is protected from modification and disclosure using DTLS.” The evaluator confirmed that all protocols listed in the TSS are specified and included in the requirements of the ST.

3.5.1.2 Guidance Documentation

258 The evaluator shall confirm that the guidance documentation contains instructions for establishing the relevant allowed communication channels and protocols between each pair of authorized TOE components, and that it contains recovery instructions should a connection be unintentionally broken.

Findings:	The DTLS-CAPWAP, FIPS mode CC Mode, and Enable LSC Provisioning for AP subsections of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] provide instructions for establishing relevant allowed communication channels and protocols between authorized TOE components. The [AGD] section “DTLS Session Interruption and Recovery” states, “If the DTLS connection used by the TOE for internal communication as specified in FPT_ITT.1. is unintentionally broken, the Security Administrator may find the AP is no longer listed in the Web GUI in the Monitoring Dashboard (Monitoring -> Wireless -> AP Statistics). If this condition occurs the AP will restart the DTLS connection and attempt to re-join the WLC automatically. No Security Administrator intervention is required for the AP to recover from an interrupted DTLS session.”

3.5.1.3 Tests

259 The evaluator shall perform the following tests:

a) Test 1: The evaluator shall ensure that communications using each protocol between each pair of authorized TOE components is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

Findings:	The TOE maintains trusted channels to TOE components via DTLS, which has been set up as per the evaluated configuration. This channel is tested throughout the evaluation. Tests 1 and 2 outlined here, are performed for the secure communication channel in the following sections: - FCS_DTLSC_EXT.1 - FCS_DTLSS_EXT.1

- FIA_X509_EXT.1/ITT
- FPT_ITT.1

b) Test 2: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.

Findings: See previous test case.

c) Test 3: Objective: The objective of this test is to ensure that the TOE reacts appropriately to any connection outage or interruption of the route between distributed components.

The evaluator shall ensure that, for each different pair of nonequivalent component types, the connection is physically interrupted for the following durations: i) a duration that exceeds the TOE's application layer timeout setting, ii) a duration that is shorter than the application layer timeout but is of sufficient length to interrupt the network link layer.

The evaluator shall ensure that when physical connectivity is restored, either communications are appropriately protected, or the secure channel is terminated and the registration process (as described in the FTP_TRP.1/Join) re-initiated, with the TOE generating adequate warnings to alert the Security Administrator.

In the case that the TOE is able to detect when the cable is removed from the device, another physical network device (e.g. a core switch) shall be used to interrupt the connection between the components.

The interruption shall not be performed at the virtual node (e.g. virtual switch) and must be physical in nature.

260 Further assurance activities are associated with the specific protocols.

Findings: This test was done as part of FTP_ITC.1 Test 4 for Distributed TOEs.

4 Evaluation Activities for NDcPP Selection-Based Requirements

4.1 Security Audit (FAU)

4.1.1 FAU_GEN_EXT.1 Security Audit Data Generation for Distributed TOE Components

261 For distributed TOEs, the requirements on TSS, Guidance Documentation and Tests regarding FAU_GEN_EXT.1 are already covered by the corresponding requirements for FAU_GEN.1.

4.1.2 FAU_STG_EXT.4 Protected Local audit event storage for distributed TOEs & FAU_STG_EXT.5 Protected Remote audit event storage for Distributed TOEs

4.1.2.1 TSS

262 The evaluator examines the TSS to confirm that it describes which TOE components store their security audit events locally and which send their security audit events to other TOE components for local storage. For the latter, the target TOE component(s) which store security audit events for other TOE components shall be identified. For every sending TOE component, the corresponding receiving TOE component(s) need to be identified. For every transfer of audit information between TOE components it shall be described how the data is secured during transfer according to FTP_ITC.1 or FPT_ITT.1.

Findings:	[ST] / TOE Summary Specification states, “the AP has transferred its contents to the WLC where it is stored locally.” [ST] / TOE Summary Specification for FAU_STG_EXT.1 states, “the AP will transmit its audit messages to the WLC over the secure DTLS channel described in FPT_ITT.1.”

263 For each TOE component which does not store audit events locally by itself, the evaluator confirms that the TSS describes how the audit information is buffered before sending to another TOE component for local storage.

Findings:	[ST] / TOE Summary Specification states, “The AP maintains the audit data in a transmission buffer and continues to do so until the AP has transferred its contents to the WLC”.

4.1.2.2 Guidance Documentation

264 The evaluator shall examine the guidance documentation to ensure that it describes how the link between different TOE components is established if audit data is exchanged between TOE components for local storage. The guidance documentation shall describe all possible configuration options for local storage of audit data and provide all instructions how to perform the related configuration of the TOE components.

Findings:	The link between TOE components and the TOE occurs over DTLS as described in the DLTS-CAPWAP subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD]. The subsection states,

	"CAPWAP is an open standard developed by the IETF for the management of wireless access points which uses DTLS to provide for secure communication between the Controller and Access Points. In this section you will configure DTLS for use by CAPWAP in order to enrol to obtain certificates for the Controller and Access Points." and "The first time an Access Point joins the Controller it must use either a manufactured-installed certificate (MIC) or a self-signed certificate (SSC). MICs and SSCs are only for the very first time the AP joins a Controller. For all subsequent joins, the AP will use Locally Significant Certificates (LSC). Locally Significant Certificates (LSCs) are obtained via Enrollment over Secure Transport (EST) and requires the organization has its own PKI and a Certificate Authority (CA) that support EST." The [AGD] does not describe any configuration of TOE component logging behaviour.

265 The evaluator shall also ensure that the guidance documentation describes for every TOE component which does not store audit information locally how audit information is buffered before transmission to other TOE components.

Findings:	[AGD] "Auditing" section states, "The AP maintains its audit data in a transmission buffer and continues to do so until the AP has transferred its contents to the WLC where it is stored locally."

4.1.2.3 Tests

266 For at least one of each type of distributed TOE components (sensors, central nodes, etc.), the following tests shall be performed using distributed TOEs.

267 Test 1: For each type of TOE component, the evaluator shall perform a representative subset of auditable actions and ensure that these actions cause the generation of appropriately formed audit records. Generation of such records can be observed directly on the distributed TOE component (if there is appropriate interface), or indirectly after transmission to a central location.

Findings:	These tests are performed throughout the evaluation.

268 Test 2: For each type of TOE component that, in the evaluated configuration, is capable of transmitting audit information to the external audit server (as specified in FTP_ITC.1), the evaluator shall configure a trusted channel and confirm that audit records generated as a result of actions taken by the evaluator are securely transmitted. It is sufficient to observe negotiation and establishment of the secure channel with the TOE component and the subsequent transmission of encrypted data to confirm this functionality. Alternatively, the following steps shall be performed: The evaluator induces audit record transmission, then reviews the packet capture around the time of transmission and verifies that no audit data is transmitted in the clear.

Findings:	These tests are satisfied by FCS_IPSEC_EXT.1.

269 Test 3: For each type of TOE component that, in the evaluated configuration, is capable of transmitting audit information to another TOE component (as specified in FTP_ITT.1 or FTP_ITC.1, respectively), the evaluator shall configure a secure channel and confirm that audit records generated as a result of actions taken by the evaluator are securely transmitted. It is sufficient to observe negotiation and establishment of the secure channel with the TOE component and the subsequent transmission of encrypted data to confirm this functionality. Alternatively, the following steps shall be performed: The evaluator induces audit record transmission, then

reviews the packet capture around the time of transmission and verifies that no audit data is transmitted in the clear.

270 While performing these tests, the evaluator shall verify that the TOE behaviour observed during testing is consistent with the descriptions provided in the TSS and the Guidance Documentation. Depending on the TOE configuration, there might be a large number of different possible configurations. In such cases, it is acceptable to perform subset testing, accompanied by an equivalency argument describing the evaluator's sampling methodology.

Findings:	These tests are satisfied by FCS_DTLSS_EXT.1 and FCS_DTLSC_EXT.1.

4.2 Cryptographic Support (FCS)

4.2.1 FCS_DTLSC_EXT.1 Extended: DTLS Client Protocol without mutual authentication

4.2.1.1 TSS

FCS_DTLSC_EXT.1.1

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

Findings:	[ST] / TOE Summary Specification specifies the following ciphersuites: "The TSF implements DTLS 1.2 conformant to RFC 6347 supporting the following ciphersuites: ■ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 ■ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 ■ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 ■ TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246 ■ TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268 ■ TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268" The evaluator confirmed that the specified ciphersuites include those listed for this component.

FCS_DTLSC_EXT.1.2

272 The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the administrator/application-configured reference identifier, including which types of reference identifiers are supported (e.g. application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported.

Findings:	[ST] / TOE Summary Specification states, "When establishing a DTLS connection, the WLC TOE Component supports a reference identifier of type id-at-

	commonName per RFC 5280 Appendix A. The AP will establish its reference identifier through a “Gatekeeper” discovery process.” The TSS states that “The TOE does not support the use of wildcards within certificates and does not support certificate pinning. Use of an IP Address reference identifier in the CN field is not supported in the evaluated configuration.”

273 Note that where a DTLS channel is being used between components of a distributed TOE for FPT_ITT.1, the requirements to have the reference identifier established by the user are relaxed and the identifier may also be established through a “Gatekeeper” discovery process. The TSS should describe the discovery process and highlight how the reference identifier is supplied to the “joining” component. Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1 and the ST author selected attributes from RFC 5280, the evaluator shall ensure the TSS describes which attribute type, or combination of attributes types, are used by the client to match the presented identifier with the configured identifier. The evaluator shall ensure the TSS presents an argument how the attribute type, or combination of attribute types, uniquely identify the remote TOE component; and the evaluator shall verify the attribute type, or combination of attribute types, is sufficient to support unique identification of the maximum supported number of TOE components.

Findings:	[ST] / TOE Summary Specification (FCS_DTLSS_EXT.1) states, “The DN is compared to the expected identifier as follows: “The CC Configuration Guide requires the Security Administrator to maintain an AP authorization list on the WLC. The AP authorization list defines the APs that are permitted to join by identification of its unique serial number. If the serial number matches the subject Distinguished Name in the certificate presented by the AP, the components will proceed to implement an internal channel protected with DTLS. If it does not match an entry in the authorization list, the DTLS internal channel will not be established and the and the AP will not be able to join.” The evaluator feels that the client using its unique serial number as its presented identifier in the DN of its presented certificate is sufficient to support unique identification of the maximum supported number of Access Points (APs).

274 If IP addresses are supported in the CN as reference identifiers, the evaluator shall ensure that the TSS describes the TOE’s conversion of the text representation of the IP address in the CN to a binary representation of the IP address in network byte order. The evaluator shall also ensure that the TSS describes whether canonical format (RFC 5952 for IPv6, RFC 3986 for IPv4) is enforced.

Findings:	The TOE does not support IP address reference identifiers in the CN.

FCS_DTLSC_EXT.1.4

275 The evaluator shall verify that TSS describes the Supported Elliptic Curves/Supported Groups Extension and whether the required behaviour is performed by default or may be configured.

Findings:	[ST] / TOE Summary Specification states, “For DTLS 1.2 connections to the WLC TOE Component, the TSF presents secp256r1, secp384r1, and secp521r1 and no other curves in the Supported Group extension of the Client Hello. This behavior is implemented by default and is not configurable.”

4.2.1.2 Guidance Documentation

FCS_DTLSC_EXT.1.1

276 The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that DTLS conforms to the description in the TSS.

Findings: Instructions on how to configure DTLS on the TOE so it conforms to the description given in the TSS is provided in the DTLS-CAPWAP, CC Mode, and Enable LSC Provisioning for AP subsections of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD].

The DTLS-CAPWAP subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] states,

"In this section you will configure DTLS for use by CAPWAP in order to enrol to obtain certificates for the Controller and Access Points."

The CC Mode subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] states,

"When obtaining a certificate for the AP to use for DTLS, the Controller must establish a mutually authenticated TLS trusted channel to the EST server. Those certificates on each side are generated by a manual out-of-band method. Once the TLS channel has been successfully established, the Controller will submit a certificate request on behalf of an Access Point to use for DTLS. The Cisco 9800 Wireless LAN Controller TOE refers to these X.509 certificates as Locally Significant Certificates (LSC).

"This section describes the configuration necessary for:

■ The Controller to obtain certificates to establish a TLS 1.2 mutually-authenticated client connection to an EST Server supporting the following ciphersuites:

…

■ The Controller and Access Point to obtain certificates to establish a DTLS 1.2 mutually-authenticated connection supporting the following ciphersuites:"

FCS_DTLSC_EXT.1.2

277 The evaluator shall ensure that the operational guidance describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE includes support for IP addresses, the evaluator shall ensure that the operational guidance provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.

Findings: The Enable LSC Provisioning for AP subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] provides instructions on configuration of the common name identifiers of peers. Support for the SAN extension is not claimed by FCS_DTLSC_EXT.1.

The Enable LSC Provisioning for AP subsections of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] states,

	"3. Configure Subject-Name Parameters in LSC Certificate (config)# ap lsc-provision subject-name country US state MA city Boxborough domain GCT org STO email-address tac@cisco.com Note: Configuration of the Common Name parameter is not required for the AP. The CN field in the certificate request is auto filled with the AP's product ID and its unique hardware serial number."

278 Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1, the SFR selects attributes from RFC 5280, and FCO_CPC_EXT.1.2 selects "no channel"; the evaluator shall verify the guidance provides instructions for establishing unique reference identifiers based on RFC5280 attributes.

Findings:	The LSC Provisioning for AP subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] provides instructions on configuration of the common name identifiers of peers. Support for the SAN extension is not claimed by FCS_DTLSC_EXT.1. The Enable LSC Provisioning for AP subsections of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] states, "3. Configure Subject-Name Parameters in LSC Certificate (config)# ap lsc-provision subject-name country US state MA city Boxborough domain GCT org STO email-address tac@cisco.com Note: Configuration of the Common Name parameter is not required for the AP. The CN field in the certificate request is auto filled with the AP's product ID and its unique hardware serial number."

FCS_DTLSC_EXT.1.4

279 If the TSS indicates that the Supported Elliptic Curves/Supported Groups Extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the Supported Elliptic Curves/Supported Groups Extension.

Findings:	The TSS does not indicate the Supported Groups Extension must be configured to meet the requirement.

4.2.1.3 Tests

280 For all tests in this chapter the DTLS server used for testing of the TOE shall be configured not to require mutual authentication.

281 For clarification: DTLS communication packets might be received in a different order than sent due to the use of the UDP protocol. All tests requiring a specific order of test steps ("before", "after") are therefore referring to the sequence numbering of DTLS packets.

FCS_DTLSC_EXT.1.1

282 Test 1: The evaluator shall establish a DTLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level application protocol, e.g., as part of a syslog session. It is sufficient to observe the successful negotiation of a ciphersuite 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 ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

283 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.

High-Level Test Description
During evaluated configuration and using a Lightship developed DTLS server, establish a DTLS connection with the TOE client using each of the ciphersuites specified by the requirement.
Findings: PASS

284 Test 2: 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.

High-Level Test Description
Construct a X.509 certificate, without the extendedKeyUsage with ‘serverAuth’. Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server and show that the handshake fails.
Findings: PASS

285 Test 3: The evaluator shall send a server certificate in the DTLS connection that does not match the server-selected ciphersuite (for example, send an ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite). The evaluator shall verify that the TOE disconnects after receiving the server’s Certificate handshake message.

High-Level Test Description
Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server using any of the claimed ciphersuites. The Lightship DTLS server will send back an otherwise validly constructed server certificate which does not match the requested the ciphersuite.
Findings: PASS

286 Test 4: The evaluator shall perform the following 'negative tests':

a) The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection.

High-Level Test Description
Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server using the TLS_NULL_WITH_NULL_NULL ciphersuite (cipher ID 0x0000).
Findings: PASS

b) Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.

High-Level Test Description
Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server sending a non-negotiated ciphersuite.
Findings: PASS

c) [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension the evaluator shall configure the server to perform an ECDHE or DHE key exchange in the DTLS connection using a non-supported curve/group (for example P-192) and shall verify that the TOE disconnects after receiving the server's Key Exchange handshake message.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server using a non-supported curve/group.
Findings: PASS

287 Test 5: The evaluator performs the following modifications to the traffic:

a) Change the DTLS version selected by the server in the Server Hello to a non-supported DTLS version and verify that the client rejects the connection.

High-Level Test Description
Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server advertising a non-supported DTLS version.
Findings: PASS

b) [conditional]: If using DHE or ECDH, modify the signature block in the Server's Key Exchange handshake message, and verify that the handshake is not finished successfully 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 DTLS, then this test shall be omitted.

High-Level Test Description
Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server. During the handshake, modify the signature block in the Server's Key Exchange handshake message.
Findings: PASS

288 Test 6: The evaluator performs the following 'scrambled message tests':

a) Modify a byte in the Server Finished handshake message and verify that the handshake is not finished successfully and no application data flows.

High-Level Test Description
Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server sending a mangled finished message.
Findings: PASS

b) Send a garbled message from the Server after the Server has issued the ChangeCipherSpec message and verify that the handshake is not finished successfully and no application data flows.

High-Level Test Description
Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server sending a mangled finished message.
Findings: PASS

c) Modify at least one byte in the server's nonce in the Server Hello handshake message and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.

High-Level Test Description
Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server sending a modified nonce value in the Server Hello handshake message.
Findings: PASS

FCS_DTLSC_EXT.1.2

289 Note that tests 1-6 are only applicable to:

a) DTLS-based trusted channel communications according to FTP_ITC.1 and trusted path communications according to FTP_TRP.1

Or:

b) DTLS-based trusted channel communications when RFC 6125 is selected for FPT_ITT.1

Findings:	The ST claims DTLS-based trusted channel communications for FPT_ITT.1. However, RFC 6125 is not selected. Therefore, tests 1-6 are not applicable.

Test 7 is only applicable to DTLS-based trusted channel communications when RFC 5280 is selected for FPT_ITT.1. Therefore, all tests are marked as conditional. Note that for some tests additional conditions apply.

290 IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:

• IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.

291 IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.

292 The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a DTLS connection:

a) Test 1 [conditional]: 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. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the CN.

Remark: 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 Not Applicable:	The ST claims DTLS-based trusted channel communications for FPT_ITT.1. However, RFC 6125 is not selected. Therefore, tests 1-6 are not applicable.

b) Test 2 [conditional]: 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 (e.g. IPv4, IPv6, FQDN, URI). When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the SAN.

Test Not Applicable:	The ST claims DTLS-based trusted channel communications for FPT_ITT.1. However, RFC 6125 is not selected. Therefore, tests 1-6 are not applicable.

c) 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. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. If the TOE does mandate the presence of the SAN extension, this test shall be omitted.

Test Not Applicable:	The ST claims DTLS-based trusted channel communications for FPT_ITT.1. However, RFC 6125 is not selected. Therefore, tests 1-6 are not applicable.

d) Test 4 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, SRV).

Test Not Applicable:	The ST claims DTLS-based trusted channel communications for FPT_ITT.1. However, RFC 6125 is not selected. Therefore, tests 1-6 are not applicable.

e) Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URIID):

1) [conditional]: 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 Not Applicable:	The ST claims DTLS-based trusted channel communications for FPT_ITT.1. However, RFC 6125 is not selected. Therefore, tests 1-6 are not applicable.

2) [conditional]: 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 succeeds, if wildcards are supported, or fails if wildcards are not supported. 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.com) and verify that the connection fails. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities).

Test Not Applicable:	The ST claims DTLS-based trusted channel communications for FPT_ITT.1. However, RFC 6125 is not selected. Therefore, tests 1-6 are not applicable.

[Updated per TD 0634]

293 Objective: The objective of this test is to ensure the TOE is able to differentiate between IP address identifiers that are not allowed to contain wildcards and other types of identifiers that may contain wildcards.

f) Test 6: [conditional] If IP address identifiers supported in the SAN or CN, the evaluator shall present a server certificate that contains a CN that matches the reference identifier, except one of the groups has been replaced with a wildcard asterisk (*) (e.g. CN=*.168.0.1 when connecting to 192.168.0.1...

This negative test corresponds to the following section of the Application Note 64/105: "The exception being, the use of wildcards is not supported when using IP address as the reference identifier."

Test Not Applicable:	The ST claims DTLS-based trusted channel communications for FPT_ITT.1. However, RFC 6125 is not selected. Therefore, tests 1-6 are not applicable.

g) Test 7:[conditional] If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator modifies each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier:

1) The evaluator shall present a server certificate that does not contain an identifier in the Subject (DN) attribute type(s) that matches the reference identifier. The evaluator shall verify that the connection fails.

High-Level Test Description
Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server. Presented server certificate should not contain an identifier in the Subject (DN) attribute type(s) that matches the reference identifier.
Findings: PASS

2) The evaluator shall present a server certificate that contains a valid identifier as an attribute type other than the expected attribute type (e.g. if the TOE is configured to expect id-at-serialNumber=correct_identifier, the certificate could instead include id-at-name=correct_identifier), and does not contain the SAN extension. The evaluator shall verify that the connection fails. Remark: 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 this test.

High-Level Test Description
Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server. Presented server certificate should contain a valid identifier as an attribute type other than the expected attribute type (i.e., OU instead of CN).
Findings: PASS

3) The evaluator shall present a server certificate that contains a Subject attribute type that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.

Findings:	TOE works trivially as it does not require the SAN. This is already tested in FCS_DTLSC_EXT.1.1 Test 1.

4) The evaluator shall confirm that all use of wildcards results in connection failure regardless of whether the wildcards are used in the left or right side of the presented identifier. (Remark: Use of wildcards is not addressed within RFC 5280.)

High-Level Test Description
Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server. Server certificate should include a wildcard in the presented identifier.
Findings: PASS

FCS_DTLSC_EXT.1.3

294 The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

295 Test 1: Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds and a trusted channel can be established.

Findings:	This is done in FIA_X509_EXT.1/Rev (RadSec) Test 1 (generic CA certificate upload) and FCS_DTLSS_EXT.1 Test 1 (presented certificate validated and trusted channel established).

296 Test 2: The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g. trusted channel was established) covering the types of failure for which an override mechanism is defined.

Findings:	Each of these failures are shown in FIA_X509_EXT.1/ITT.

297 Test 3 [conditional]: The purpose of this test to verify that only selected certificate validation failures could be administratively overridden. If any override mechanism is defined for failed certificate validation, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative override available to accept such certificate.

Findings:	No such overrides are claimed.

FCS_DTLSC_EXT.1.4

298 Test 1 [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension, the evaluator shall configure the server to perform ECDHE or DHE (as applicable) 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.

High-Level Test Description
Using a Lightship developed DTLS server, force the TOE client to attempt a handshake with a test server. For each of attempt, the server should perform a key exchange using each of the TOE's supported curves and/or groups.
Findings: PASS

4.2.2 FCS_DTLSS_EXT.1 Extended: DTLS Server Protocol without mutual authentication

4.2.2.1 TSS

FCS_DTLSS_EXT.1.1

299 The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component.

Findings: [ST] / TOE Summary Specification specifies the following ciphersuites:

"The TSF implements DTLS 1.2 conformant to RFC 6347 supporting the following ciphersuites:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268"

The evaluator confirmed that the specified ciphersuites are identical to those listed for this component.

FCS_DTLSS_EXT.1.3

300 The evaluator shall verify that the TSS describes how the DTLS Client IP address is validated prior to issuing a ServerHello message.

Findings: [ST] / TOE Summary Specification states, "Upon receiving the Client Hello message the WLC sends a Hello Verify Request message and performs a stateless cookie exchange to ensure the DTLS Client is not being spoofed."

FCS_DTLSS_EXT.1.4

301 If using ECDHE or DHE ciphers, the evaluator shall verify that the TSS describes the key agreement parameters of the server Key Exchange message.

Findings:	[ST] / TOE Summary Specification states, “DTLS Server key establishment is implemented as follows: ■ If DHE_RSA_* ciphersuites are configured, the WLC generates the Diffie-Hellman 2048 bit ephemeral key agreement parameters, prime ‘p’ and generator ‘g’ which has at a minimum a 112-bit level of security. The prime ‘p’ and generator ‘g’ parameters are transmitted to the client in the Server Key Exchange message on each connection attempt. ■ If TLS_ECDHE_* ciphersuites are configured, the secp384r1 NIST elliptic curve will be used by default. On each connection attempt, the Server Key Exchange message includes: 1) the NIST named curve which specifies predefined EC domain parameters; and 2) an ECDH public key corresponding to those parameters.“

FCS_DTLSS_EXT.1.5

302 The evaluator shall verify that the TSS describes the actions that take place if a message received from the DTLS Client fails the MAC integrity check.

Findings:	[ST] / TOE Summary Specification states, “if there is a Message Authentication Code (MAC) verification failure, the WLC will silently discard the record and continue with the connection. The WLC will increment its DTLS packet error counter.”

FCS_DTLSS_EXT.1.6

303 The evaluator shall verify that TSS describes how replay is detected and silently discarded for DTLS records that have previously been received and too old to fit in the sliding window.

Findings:	[ST] / TOE Summary Specification states, “The WLC enforces replay detection using sequence numbers. Valid record sequence numbers are maintained in a sliding window. For each record received, the TOE verifies if it is in the window boundary. Messages that are received where the same record was previously received or too old to fit in the sliding window are silently discarded.”

FCS_DTLSS_EXT.1.7

304 The evaluator shall verify that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246) and/or if session resumption based on session tickets is supported (RFC 5077).

Findings:	[ST] / TOE Summary Specification states, “The TSF implements support for session resumption based on session IDs according to RFC 5246 (TLS1.2) using multiple contexts.”

305 If session tickets are supported, the evaluator shall verify that the TSS describes that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption. The evaluator shall verify that the TSS identifies the key lengths and algorithms used to protect session tickets.

Findings:	The ST does not claim support to session resumption based on session tickets.

306 If session tickets are supported, the evaluator shall verify that the TSS describes that session tickets adhere to the structural format provided in section 4 of RFC 5077 and if not, a justification shall be given of the actual session ticket format.

Findings:	The ST does not claim support to session resumption based on session tickets.

Note:	Updated per TD0569.

307 If the TOE claims a (D)TLS server capable of session resumption (as a single context, or across multiple contexts), the evaluator verifies that the TSS describes how session resumption operates (i.e. what would trigger a full handshake, e.g. checking session status, checking Session ID, etc.). If multiple contexts are used the TSS describes how session resumption is coordinated across those contexts. In case session establishment and session resumption are always using a separate context, the TSS shall describe how the contexts interact with respect to session resumption (in particular regarding the session ID). It is acceptable for sessions established in one context to be resumable in another context.

Findings:	[ST] / TOE Summary Specification states, "The contexts are coordinated as follows: After the WLC successfully authenticates the AP, an internal trusted channel is established. This control channel protects the management traffic between a WLC and AP. When Enable Data DTLS is configured, as instructed in the Common Criteria Configuration Guide, a second channel is established using TLS session resumption. This data channel protects user data sent from the wireless client destined to the VLAN on the wired interface. The session resumption functions as follows: If the WLC determines there is a session ID match with the AP and the control channel session state is still valid, the WLC will proceed with an abbreviated handshake and send a Server Hello message with the matched Session ID. Both AP and WLC will then exchange ChangeCipherSpec and Finished messages. If the WLC determines there is not a Session ID match with the AP, the WLC requires a full TLS handshake to establish the data channel."

4.2.2.2 Guidance Documentation

FCS_DTLSS_EXT.1.1

308 The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that DTLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements).

Findings:	The DTLS-CAPWAP, CC Mode and Enable LSC Provisioning for AP subsections of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] provide instructions on configuring the TOE so that DTLS conforms to the description given in the TSS. The Enable LSC Provisioning for AP subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] specifically provides instructions on configuring supported ciphersuites so the toe conforms to the description given in the TSS.

FCS_DTLSS_EXT.1.4

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

Findings:	The [AGD] does not identify any necessary additional configurations to meet the key establishment requirements of FCS_DTLSS_EXT.1.4.

NOTE:	Updated per TD0569.

FCS_DTLSS_EXT.1.7

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

Findings:	The [AGD] does not identify any necessary additional configurations to meet the session resumption requirements of FCS_DTLSS_EXT.1.7.

4.2.2.3 Tests

For clarification: For DTLS communication packets might be received in a different order than sent due to the use of the UDP protocol. All tests requiring a specific order of test steps ("before", "after") are therefore referring to the sequence numbering of DTLS packets.

FCS_DTLSS_EXT.1.1

311 Test 1: The evaluator shall establish a DTLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level application protocol, e.g., as part of a syslog session. It is sufficient to observe the successful negotiation of a ciphersuite 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 ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

High-Level Test Description
Using the Lightship Security DTLS client test tool as a DTLS client, initiate a DTLS connection to the WLC using each claimed ciphersuite. Observe the successful completion of the DTLS handshake with the WLC. Note that the wireless management trustpoint needs to be RSA-based for RSA ciphersuite tests and ECC-based for ECDSA ciphersuite tests. (See above command)
Findings: PASS

312 Test 2: The evaluator shall send a Client Hello to the server with a list of ciphersuites that does not contain any of the ciphersuites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the server denies the connection.

High-Level Test Description
Using the Lightship Security DTLS client test tool as a DTLS client, initiate a DTLS connection to the WLC using the following ciphersuites: • TLS_RSA_WITH_NULL_MD5 • TLS_NULL_WITH_NULL_NULL Verify the server denies the connection.
Findings: PASS

313 Test 3: The evaluator shall perform the following modifications to the traffic:

a) Modify a byte in the Client Finished handshake message and verify that the server rejects the connection and does not send any application data.

High-Level Test Description
Using the Lightship security DTLS client test tool as a DTLS client, initiate a DTLS connection to the WLC and send a Client Finished handshake message with a modified byte. Verify that the server rejects the connection and does not send any application data.
Findings: PASS

b) (Test Intent: The intent of this test is to ensure that the server's TLS implementation immediately makes use of the key exchange and authentication algorithms to: a) Correctly encrypt (D)TLS Finished message and b) Encrypt every (D)TLS message after session keys are negotiated.)

The evaluator shall use one of the claimed ciphersuites to complete a successful handshake and observe transmission of properly encrypted application data. The evaluator shall verify that no Alert with alert level Fatal (2) messages were sent.

The evaluator shall verify that the Finished message (Content type hexadecimal 16 and handshake message type hexadecimal 14) is sent immediately after the server's ChangeCipherSpec (Content type hexadecimal 14) message. The evaluator shall examine the Finished message (encrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 11 22 33 44 55...) and confirm that it does not contain unencrypted data (unencrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 14 00 00 0c...), by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14 for at least one of three test messages. There is a chance that an encrypted Finished message contains a hexadecimal value of '14' at the position where a plaintext Finished message would contain the message type code '14'. If the observed Finished message contains a hexadecimal value of '14' at the position where the plaintext Finished message would contain the message type code, the test shall be repeated three times in total. In case the value of '14' can be observed in all three tests it can be assumed that the Finished message has indeed been sent in plaintext and the test has to be regarded as 'failed'. Otherwise it has to be assumed that the observation of the value '14' has been due to chance and that the Finished message has indeed been sent encrypted. In that latter case the test shall be regarded as 'passed'.

High-Level Test Description
Using the Lightship security DTLS client test tool as a DTLS client, initiate a normal DTLS connection to the WLC. Verify the Encrypted Handshake Message does not have a 0x14 in the position indicated by the test to show it is encrypted and that connection succeeds.
Findings: PASS

FCS_DTLSS_EXT.1.3

314 Modify at least one byte in the cookie from the Server's HelloVerifyRequest message and verify that the Server rejects the Client's handshake message.

High-Level Test Description
Using the Lightship security DTLS client test tool as a DTLS client, initiate a DTLS connection to the WLC and send a Client Hello with a modified version of the cookie sent of the HelloVerifyRequest sent by the WLC. Verify that the server rejects the connection and does not send any application data.

High-Level Test Description
Findings: PASS

FCS_DTLSS_EXT.1.4

315 Test 1 [conditional]: If ECDHE ciphersuites are supported:

a) The evaluator shall repeat this test for each supported elliptic curve. The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single supported elliptic curve specified in the Elliptic Curves Extension. The evaluator shall verify (through a packet capture or instrumented client) that the TOE selects the same curve in the Server Key Exchange message and successfully establishes the connection.

b) The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single unsupported elliptic curve (e.g. secp192r1 (0x13)) specified in RFC4492, chap. 5.1.1. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.

High-Level Test Description
Using the Lightship security DTLS client test tool as a DTLS client, initiate a DTLS connection to the WLC using the specified ECDHE ciphersuite and a supported elliptic curve within the Elliptic Curves Extension. Verify that the TOE selects the same supported curve in its Server Key Exchange message and the connection succeeds. Using the Lightship security DTLS client test tool as a DTLS client, initiate a DTLS connection to the WLC using a supported ECDHE ciphersuite and an unsupported elliptic curve within the Elliptic Curves Extension. Verify that the TOE does not send a Server Hello message and the connection is not successfully established.
Findings: PASS

316 Test 2 [conditional]: If DHE ciphersuites are supported, the evaluator shall repeat the following test for each supported parameter size. If any configuration is necessary, the evaluator shall configure the TOE to use a supported Diffie-Hellman parameter size. The evaluator shall attempt a connection using a supported DHE ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Exchange Message where p Length is consistent with the configured Diffie-Hellman parameter size(s).

High-Level Test Description
Using the Lightship security DTLS client test tool as a DTLS client, initiate a DTLS connection to the WLC using the specified Diffie-Hellman ciphersuite. Verify that the TOE sends a p Length in its Server Key Exchange message consistent with the supported DH parameter size and that the connection succeeds
Findings: PASS

317 Test 3 [conditional]: If RSA key establishment ciphersuites are supported, the evaluator shall repeat this test for each RSA key establishment key size. If any configuration is necessary, the evaluator shall configure the TOE to perform RSA key establishment using a supported key size (e.g. by loading a certificate with the

appropriate key size). The evaluator shall attempt a connection using a supported RSA key establishment ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a certificate whose modulus is consistent with the configured RSA key size.

Test Not Applicable:	The TOE does not claim support for RSA key establishment ciphersuites for DTLS.

FCS_DTLSS_EXT.1.5

318 The evaluator shall establish a connection using a client. The evaluator will then modify at least one byte in a record message and verify that the Server discards the record or terminates the DTLS session.

High-Level Test Description
Using the Lightship security DTLS client test tool as a DTLS client, initiate a DTLS connection to the WLC and subsequently send modified record messages to the TOE. Verify that the TOE discards the record or terminates the DTLS session.
Findings: PASS

FCS_DTLSS_EXT.1.6

319 The evaluator shall set up a DTLS connection. The evaluator shall then capture traffic sent from the DTLS Client to the TOE. The evaluator shall retransmit copies of this traffic to the TOE in order to impersonate the DTLS Client. The evaluator shall observe that the TSF does not take action in response to receiving these packets and that the audit log indicates that the replayed traffic was discarded.

High-Level Test Description
Using a custom tool, capture legitimate DTLS traffic between the WLC and an Access Point and replay copies of this DTLS traffic from another host to the WLC, attempting to impersonate the DTLS client. Verify the TSF does not take action in response to the replayed packets and that the audit log indicates the replayed traffic was discarded.
Findings: PASS

FCS_DTLSS_EXT.1.7

320 Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption)

321 Test 1 [conditional]: If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test:

a) The client sends a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket.

b) The client verifies the server does not send a NewSessionTicket handshake message (at any point in the handshake).

c) The client verifies the Server Hello message contains a zero-length session identifier or passes the following steps:

Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID.

d) The client completes the TLS handshake and captures the SessionID from the ServerHello.

e) The client sends a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session from step d) open or by starting a new TLS session using the SessionID captured in step d).

f) The client verifies the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

NOTE:	Updated per TD0569.

322 Remark: If multiple contexts are supported for session resumption, the session ID or session ticket may be obtained in one context for resumption in another context. It is possible that one or more contexts may only permit the construction of sessions to be reused in other contexts but not actually permit resumption themselves. For contexts which do not permit resumption, the evaluator is required to verify this behaviour subject to the description provided in the TSS. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

Test Not Applicable:	The TOE supports session resumption based on session IDs according to RFC 5246 (TLS1.2).

323 Test 2 [conditional]: If the TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

a) The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in figure 2 of RFC 4346 or RFC 5246).

High-Level Test Description
Using the Lightship Security test DTLS Session ID tool, listen for a newly established handshake message on the DTLS control channel, capture the Session ID and other relevant information and use this information to initiate an abbreviated DTLS handshake on the DTLS data channel. Verify the TOE responds with the same Session ID in its Server Hello (sent on the data channel) and that a ChangeCipherSpec and Finished message immediately follow the Server Hello as shown in Figure 3 of RFC5246.
Findings: PASS

324 b) The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

NOTE:	Updated per TD0569.

325 Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ID may be obtained in one context for resumption in another context. There is no requirement that the session ID be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ID constructed in another context must be tested. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

High-Level Test Description
Using the Lightship Security DTLS client test tool, initiate a DTLS handshake to the WLC. Capture the Session ID sent in the Server Hello and subsequently send a fatal alert message to disrupt the handshake. Attempt to start an abbreviated DTLS handshake using the previously captured Session ID. Verify the TOE implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246).
Findings: PASS

326 Test 3 [conditional]: If the TOE supports session tickets according to RFC5077, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

a) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE responds with a ServerHello with an empty SessionTicket extension, NewSessionTicket, ChangeCipherSpec and Finished messages (as seen in figure 2 of RFC 5077).

b) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data.

NOTE:	Updated per TD0569.

327 Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ticket may be obtained in one context for resumption in another context. There is no requirement that the session ticket be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ticket constructed in another context must be tested. It is not mandated that the session establishment and session resumption

share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

Test Not Applicable:	The TOE only supports session resumption based on session IDs according to RFC 5246 (TLS1.2).

4.2.3 FCS_HTTPS_EXT.1 HTTPS Protocol

4.2.3.1 TSS

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

Findings:	[ST] / TOE Summary Specification states, “The TSF implements HTTPS conformant to RFC 2818 to provide a secure interactive Web interface for remote administrative functions. The TLS Server implementation is conformant to RFC 5246”. The evaluator determined that this complies with RFC 2818.

4.2.3.2 Guidance Documentation

329 The evaluator shall examine the guidance documentation to verify it instructs the Administrator how to configure TOE for use as an HTTPS client or HTTPS server.

Findings:	The HTTPS subsection of the Remote Administration Protocols subsection of the section Preparative Procedures and Operational Guidance for the TOE of the [AGD] provides instructions on how to configure the TOE for use as an HTTPS server. The subsection states, “HTTPS is used by the Administrator to securely access the WebGUI from a remote workstation. The steps below provide instructions to configure HTTPS.”

4.2.3.3 Tests

330 This test is now performed as part of FIA_X509_EXT.1/Rev testing.

331 Tests are performed in conjunction with the TLS evaluation activities.

332 If the TOE is an HTTPS client or an HTTPS server utilizing X.509 client authentication, then the certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.

4.2.4 FCS_IPSEC_EXT.1 IPsec Protocol

4.2.4.1 TSS

FCS_IPSEC_EXT.1.1

333 The evaluator shall examine the TSS and determine that it describes what takes place when a packet is processed by the TOE, e.g., the algorithm used to process the packet. The TSS describes how the SPD is implemented and the rules for processing both inbound and outbound packets in terms of the IPsec policy. The TSS describes

the rules that are available and the resulting actions available after matching a rule. The TSS describes how those rules and actions form the SPD in terms of the BYPASS (e.g., no encryption), DISCARD (e.g., drop the packet), and PROTECT (e.g., encrypt the packet) actions defined in RFC 4301.

334 As noted in section 4.4.1 of RFC 4301, the processing of entries in the SPD is non-trivial and the evaluator shall determine that the description in the TSS is sufficient to determine which rules will be applied given the rule structure implemented by the TOE. For example, if the TOE allows specification of ranges, conditional rules, etc., the evaluator shall determine that the description of rule processing (for both inbound and outbound packets) is sufficient to determine the action that will be applied, especially in the case where two different rules may apply. This description shall cover both the initial packets (that is, no SA is established on the interface or for that particular packet) as well as packets that are part of an established SA.

Findings:	[ST] / TOE Summary Specification states, "The traffic matching the permit ACLs would then flow through the IPsec tunnel and be classified as "PROTECTED". Traffic that does not match a permit ACL in the crypto map, but that is not disallowed by other ACLs on the interface is allowed to BYPASS the tunnel. Traffic that does not match a permit ACL and is also blocked by other non-crypto ACLs on the interface would be DISCARDED. Rules applied to an access control list can be applied to either inbound or outbound traffic."

FCS_IPSEC_EXT.1.3

335 The evaluator checks the TSS to ensure it states that the VPN can be established to operate in transport mode and/or tunnel mode (as identified in FCS_IPSEC_EXT.1.3).

Findings:	[ST] / TOE Summary Specification states, "The TOE provides IPsec protection supporting one of two modes: 1) With a syslog server operating as an IPsec peer of the TOE (transport mode); or 2) With a syslog server is not directly co-located with the TOE but is adjacent to an IPsec peer within a trusted facility, and the syslog records are tunnelled over the public network (tunnel mode)."

FCS_IPSEC_EXT.1.4

336 The evaluator shall examine the TSS to verify that the selected algorithms are implemented. In addition, the evaluator ensures that the SHA-based HMAC algorithm conforms to the algorithms specified in FCS_COP.1/KeyedHash Cryptographic Operations (for keyed-hash message authentication) and if the SHA-based HMAC function truncated output is utilized it must also be described.

Findings:	[ST] / TOE Summary Specification identifies the following algorithms and no SHA-based HMAC algorithms: ■ AES-GCM-128 and AES-GCM-256 There were no SHA-based HMAC algorithms identified.

FCS_IPSEC_EXT.1.5

337 The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented.

338 For IKEv1 implementations, the evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. It may be that this is a configurable option.

Findings:	[ST] / TOE Summary Specification states, “The TOE supports IKEv2 session establishment.” The TOE does not claim IKEv1 implementation.

FCS_IPSEC_EXT.1.6

339 The evaluator shall ensure the TSS identifies the algorithms used for encrypting the IKEv1 and/or IKEv2 payload, and that the algorithms chosen in the selection of the requirement are included in the TSS discussion.

Findings:	[ST] / TOE Summary Specification identifies the following algorithms: ■ AES-GCM-128 and AES-GCM-256 The evaluator confirmed that they conform to the algorithms selected in FCS_IPSEC_EXT.1.6.

FCS_IPSEC_EXT.1.7

340 The evaluator shall ensure the TSS identifies the lifetime configuration method used for limiting the IKEv1 Phase 1 SA lifetime and/or the IKEv2 SA lifetime. The evaluator shall verify that the selection made here corresponds to the selection in FCS_IPSEC_EXT.1.5.

Findings:	[ST] / TOE Summary Specification states, “The time values for Phase 1 SAs can be limited up to 24 hours” This corresponds to the selection in FCS_IPSEC_EXT.1.7.

FCS_IPSEC_EXT.1.8

341 The evaluator shall ensure the TSS identifies the lifetime configuration method used for limiting the IKEv1 Phase 2 SA lifetime and/or the IKEv2 Child SA lifetime. The evaluator shall verify that the selection made here corresponds to the selection in FCS_IPSEC_EXT.1.5.

Findings:	[ST] / TOE Summary Specification **-** “The time values for Phase 1 SAs can be limited up to 24 hours and for Phase 2 SAs up to 8 hours.” This corresponds to the selection in FCS_IPSEC_EXT.1.8.

FCS_IPSEC_EXT.1.9

342 The evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating "x". The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of "x" meets the stipulations in the requirement.

Findings:	[ST] / TOE Summary Specification states, “The TSF generates the secret value 'x' used in the IKEv2 Diffie-Hellman key exchange ('x' in g^x mod p) using the NIST approved DRBG specified in FCS_RBG_EXT.1 and having possible lengths of 256 or 384 bits. When a random number is needed for a nonce, the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^{128}. The nonce is likewise generated using the AES-CTR DRBG.”

FCS_IPSEC_EXT.1.10

343 If the first selection is chosen, the evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of the nonces meet the stipulations in the requirement.

Findings:	The first selection was not chosen.

344 If the second selection is chosen, the evaluator shall check to ensure that, for each PRF hash supported, the TSS describes the process for generating each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of the nonces meet the stipulations in the requirement.

Findings:	[ST] / TOE Summary Specification states, “The nonce is likewise generated using the AES-CTR DRBG.” “The length of the nonce is equal to that of the hash PRF used in the session establishment (for SHA-256 hash based PRF the nonce is 256-bits and for SHA-384 Hash based PRF the nonce is 384-bits).” This satisfies the requirements in this PP and the length of the nonces meet the requirements of the ST of 128 bits and at least half the output size of the PRF hash.

FCS_IPSEC_EXT.1.11

345 The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks to ensure the TSS describes how a particular DH group is specified/negotiated with a peer.

Findings:	[ST] / TOE Summary Specification states, “The TOE supports Diffie-Hellman Groups 19 and 20.” This is consistent with the DH groups specified in the requirement. “Each IKE negotiation begins by agreement of both peers on a common (shared) IKE policy. This policy states which security parameters will be used to protect subsequent IKE negotiations and mandates how the peers are authenticated.” “The Security Administrator can configure multiple, prioritized policies on each peer, each with a different combination of parameter values. However, at least one of these policies must contain exactly the same encryption, hash, authentication, and Diffie-Hellman parameter values as one of the policies on the remote peer.”

FCS_IPSEC_EXT.1.12

346 The evaluator shall check that the TSS describes the potential strengths (in terms of the number of bits in the symmetric key) of the algorithms that are allowed for the IKE and ESP exchanges. The TSS shall also describe the checks that are done when negotiating IKEv1 Phase 2 and/or IKEv2 CHILD_SA suites to ensure that the strength (in terms of the number of bits of key in the symmetric algorithm) of the negotiated algorithm is less than or equal to that of the IKE SA this is protecting the negotiation.

Findings:	[ST] / TOE Summary Specification states, “The resulting potential strength of the symmetric key will be 128 or 256 bits of security depending on the algorithms negotiated between the two IPsec peers. As part of this negotiation, the TOE verifies that the negotiated phase 2 symmetric algorithm key strength is at most as large as the negotiated phase 1 key strength as configured on the TOE and peer via an explicit check.”

FCS_IPSEC_EXT.1.13

347 The evaluator ensures that the TSS identifies RSA and/or ECDSA as being used to perform peer authentication. The description must be consistent with the algorithms as specified in FCS_COP.1/SigGen Cryptographic Operations (for cryptographic signature).

Findings:	[ST] / TOE Summary Specification – "The TOE supports authentication of IPsec peers using pre-shared keys, and ECDSA or RSA X.509 certificates." This is consistent with the algorithms specified in FCS_COP.1/SigGen.

348 If pre-shared keys are chosen in the selection, the evaluator shall check to ensure that the TSS describes how pre-shared keys are established and used in authentication of IPsec connections. The description in the TSS shall also indicate how pre-shared key establishment is accomplished for TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key.

Findings:	[ST] / TOE Summary Specification – "Pre-shared keys must be entered my the Security Administrator and must be of length 22 characters or greater. During IKE establishment, IPsec peers authenticate each other by creating and exchanging a hash value that includes the pre-shared key. The TOE will compare the received hash value to its computed hash and determine if it matches. If it does, pre-shared key authentication is successful; otherwise pre-shared key authentication fails."

FCS_IPSEC_EXT.1.14

349 The evaluator shall ensure that the TSS describes how the TOE compares the peer's presented identifier to the reference identifier. This description shall include which field(s) of the certificate are used as the presented identifier (DN, Common Name, or SAN). If the TOE simultaneously supports the same identifier type in the CN and SAN, the TSS shall describe how the TOE prioritizes the comparisons (e.g. the result of comparison if CN matches but SAN does not). If the location (e.g. CN or SAN) of non-DN identifier types must explicitly be configured as part of the reference identifier, the TSS shall state this. If the ST author assigned an additional identifier type, the TSS description shall also include a description of that type and the method by which that type is compared to the peer's presented certificate, including what field(s) are compared and which fields take precedence in the comparison.

Findings:	[ST] / TOE Summary Specification states, "The Administrator is instructed in the CC Configuration Guide to specify one or more certificate fields together with their matching criteria and the value to match. In the evaluated configuration, the field name must specify the SAN (alt-subject-name) or CN (subject-name) fields. Match criteria should be "eq" for equal." "SAN example: alt-subject-name eq <peer.cisco.com> CN example: subject-name eq <peer>" "The TOE will reject the IKE connection in any of these situations: 1) If the data ID Payload for any of those ID Types does not match the peer's certificate exactly; 2) If an ID Payload is not provided by the peer; 3) If multiple ID Types are provided in the ID Payload."

4.2.4.2 Guidance Documentation

FCS_IPSEC_EXT.1.1

350 The evaluator shall examine the guidance documentation to verify it instructs the Administrator how to construct entries into the SPD that specify a rule for processing a packet. The description includes all three cases – a rule that ensures packets are encrypted/decrypted, dropped, and flow through the TOE without being encrypted. The evaluator shall determine that the description in the guidance documentation is consistent with the description in the TSS, and that the level of detail in the guidance documentation is sufficient to allow the administrator to set up the SPD in an unambiguous fashion. This includes a discussion of how ordering of rules impacts the processing of an IP packet.

Findings:	The IPsec subsection of the section, Preparative Procedures and Operational Guidance for the TOE in the [AGD] provides instructions on how to configure entries in the SPD that specify rules for processing packets. On the TOE, this is accomplished through Crypto Maps and associated Access Control Lists (ACLs). Furthermore, the Security Policy Database (SPD) subsection of the IPsec subsection provides a description of “PROTECTED” (encrypted), “DISCARD” (dropped) and “BYPASS” (flow through) traffic and how traffic is categorized according to the Crypto Maps and ACLs configured on the TOE. This description is consistent with the one given in the TSS and gives sufficient detail to allow an administrator to set up the SPD in an unambiguous fashion.

FCS_IPSEC_EXT.1.3

351 The evaluator shall confirm that the guidance documentation contains instructions on how to configure the connection in each mode selected.

Findings:	Instructions on how to configure transport and tunnel modes for IPsec are provided in the IPsec subsection of the section Preparative Procedures and Operational Guidance for the TOE in the [AGD]. The subsection states, “2. Define the IPsec mode which is either tunnel mode or transport mode. WLC(cfg-crypto-trans)# mode <transport | tunnel>”

FCS_IPSEC_EXT.1.4

352 The evaluator checks the guidance documentation to ensure it provides instructions on how to configure the TOE to use the algorithms selected.

Findings:	Instructions on how to configure IPsec ESP encryption and hashing algorithms are provided in the IPsec subsection of the section Preparative Procedures and Operational Guidance for the TOE in the [AGD]. The subsection states, “c. Set the encryption algorithm(s) for the proposal. WLC(config-ikev2-proposal)# encryption < aes-gcm-128 | aes-gcm-256>”

FCS_IPSEC_EXT.1.5

353 The evaluator shall check the guidance documentation to ensure it instructs the administrator how to configure the TOE to use IKEv1 and/or IKEv2 (as selected), and how to configure the TOE to perform NAT traversal (if selected).

Findings:	Instructions on how to configure IKEv2 are provided in the IPsec subsection of the section Preparative Procedures and Operational Guidance for the TOE in the [AGD]. The TOE does not claim NAT traversal and IKEv1 is not supported.

354 If the IKEv1 Phase 1 mode requires configuration of the TOE prior to its operation, the evaluator shall check the guidance documentation to ensure that instructions for this configuration are contained within that guidance.

Findings:	The TOE does not claim IKEv1 functionality.

FCS_IPSEC_EXT.1.6

355 The evaluator ensures that the guidance documentation describes the configuration of all selected algorithms in the requirement.

Findings:	The [AGD] does not identify any necessary configuration to ensure the selected cryptographic algorithms are used to encrypt IKEv2 payloads.

FCS_IPSEC_EXT.1.7

[Updated per TD 0633]

356 The evaluator shall verify that the values for SA lifetimes can be configured and that the instructions for doing so are located in the guidance documentation. If time-based limits are supported, configuring the limit may lead to a rekey no later than the specified limit. For some implementations, it may be necessary, though, to configure the TOE with a lower time value to ensure a rekey is performed before the maximum SA lifetime of 24 hours is exceeded (e.g. configure a time value of 23h 45min to ensure the actual rekey is performed no later than 24h). The evaluator shall verify that the guidance documentation allows the Administrator to configure the Phase 1 SA value of 24 hours or provides sufficient instruction about the time value to configure to ensure the rekey is performed no later than the maximum SA lifetime of 24 hours. It is not permitted to configure a value of 24 hours if that leads to an actual rekey after more than 24hours. Currently there are no values mandated for the number of bytes, the evaluator just ensures that this can be configured if selected in the requirement.

Findings:	Instructions on how to configure Phase 1 SA lifetimes are provided in the IPsec subsection of the section Preparative Procedures and Operational Guidance for the TOE in the [AGD]. Lifetimes can be configured using time limits between 2 minutes and 24 hours. The subsection states, “f. Set the IKE SA lifetime in seconds. WLC(config-ikev2-profile)# lifetime <120-86400>”

FCS_IPSEC_EXT.1.8

[Updated per TD 0633]

357 The evaluator shall verify that the values for SA lifetimes can be configured and that the instructions for doing so are located in the guidance documentation. If time-based limits are supported, configuring the limit may lead to a rekey no later than the specified limit. For some implementations, it may be necessary, though, to configure the TOE with a lower time value to ensure a rekey is performed before the maximum SA lifetime of 8 hours is exceeded (e.g. configure a time value of 7h 45min to ensure the actual rekey is performed no later than 8h). The evaluator shall verify that the guidance documentation allows the Administrator to configure the Phase 2 SA value of 8 hours or provides sufficient instruction about the time value to configure to ensure the rekey is performed no later than the maximum SA lifetime of 8 hours. It is not permitted to configure a value of 8 hours if that leads to an actual rekey after more than 8hours. Currently there are no values mandated for the number of bytes, the evaluator just ensures that this can be configured if selected in the requirement.

Findings:	Instructions on how to configure Phase 2 SA lifetimes are provided in the IPsec subsection of the section Preparative Procedures and Operational Guidance for the TOE in the [AGD]. Lifetimes can be configured using time limits between 2 minutes and 8 hours or using data-based limits.

	The subsection states, "4. Define the IPsec security association lifetime. The lifetime can be chosen based on time (hours) or can be volume based. A time-based lifetime must be entered in seconds where 1 hour=3600 seconds and 8 hours=28800 seconds. WLC(config)# crypto ipsec security-association lifetime <seconds < 120-28800>> | <kilobytes <2560-4294967295>>"

FCS_IPSEC_EXT.1.11

358 The evaluator ensures that the guidance documentation describes the configuration of all algorithms selected in the requirement.

Findings:	Instructions on how to configure DH group 19 and 20 are provided in the IPsec subsection of the section Preparative Procedures and Operational Guidance for the TOE in the [AGD]. These algorithms are consistent with the selection made in the requirement. The subsection states, "e. Set the Diffie-Hellman group(s) WLC(config-ikev2-proposal)# group <19 | 20>"

FCS_IPSEC_EXT.1.13

359 The evaluator ensures the guidance documentation describes how to set up the TOE to use certificates with RSA and/or ECDSA signatures and public keys.

Findings:	Instructions on how to configure certificates with RSA and ECDSA signatures and public keys are provided in the IPsec subsection of the section Preparative Procedures and Operational Guidance for the TOE in the [AGD].

360 The evaluator shall check that the guidance documentation describes how pre-shared keys are to be generated and established. The description in the guidance documentation shall also indicate how pre-shared key establishment is accomplished for TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key.

Findings:	Instructions on how to configure pre-shred keys are provided in the IPsec subsection of the section Preparative Procedures and Operational Guidance for the TOE in the [AGD]. Pre-shared keys are manually configured by an administrator according to the provided instructions in the [AGD]. The subsection states, "3. Configure the IKEv2 Keyring. If you chose pre-shared key as the authentication method you must complete these steps."

361 The evaluator will ensure that the guidance documentation describes how to configure the TOE to connect to a trusted CA and ensure a valid certificate for that CA is loaded into the TOE and marked "trusted".

Findings:	Instructions on how to configure the TOE to connect to a trusted CA and ensue a valid certificate for that CA is loaded into the TOE and marked "trusted" are provided in the IPsec subsection of the section Preparative Procedures and Operational Guidance for the TOE in the [AGD].

	The subsection states, “IPsec must be configured to use X.509v3 certificates supporting a minimum path length of three (root CA -> intermediate CA -> end-entity). Therefore, you will need to create two trustpoints. The section below provides steps to create a root CA and a subordinate CA using CA certificates from your organization’s PKI. Before proceeding, please have the root CA and subordinate CA certificates ready for import from your CA administrator”

FCS_IPSEC_EXT.1.14

362 The evaluator shall ensure that the operational guidance describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not, and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE does not guarantee unique identifiers, the evaluator shall ensure that the operational guidance provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.

Findings:	A description of the supported identifiers and how to configure them for IPsec peer identity checking is provided in the IPsec subsection of the section Preparative Procedures and Operational Guidance for the TOE in the [AGD]. The subsection states, “2. Specify one or more certificate fields together with their matching criteria and the value to match. In the evaluated configuration, the field name must specify the SAN (alt-subject-name) field of the peer’s certificate. Match criteria should be “eq” for equal. For example: WLC(ca-certificate-map)# alt-subject-name eq < peer.cisco.com>”

4.2.4.3 Tests

FCS_IPSEC_EXT.1.1

363 The evaluator uses the guidance documentation to configure the TOE to carry out the following tests:

a. Test 1: The evaluator shall configure the SPD such that there is a rule for dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext. The selectors used in the construction of the rule shall be different such that the evaluator can generate a packet and send packets to the gateway with the appropriate fields (fields that are used by the rule - e.g., the IP addresses, TCP/UDP ports) in the packet header. The evaluator performs both positive and negative test cases for each type of rule (e.g. a packet that matches the rule and another that does not match the rule). The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behaviour: appropriate packets were dropped, allowed to flow without modification, encrypted by the IPsec implementation.

High-Level Test Description
The TOE implements an SPD through the use of Access Control Lists and Crypto Maps. An ACL may be associated with an IPsec Crypto Map, which is subsequently associated with a (virtual) interface. The following rules apply to traffic traversing the TOE.

High-Level Test Description
• Traffic matching a permit statement within an ACL associated with the IPSec Crypto Map is encrypted and flows through the IPsec tunnel; • Unencrypted traffic that matches a permit statement within an ACL associated with the IPSec Crypto Map is dropped; • Traffic that does not match a permit statement within an ACL associated with the IPSec Crypto Map but is not disallowed by another ACL will flow in plaintext, bypassing the IPSec tunnel. Establish the IPsec VPN connection. Create an ACL to send ICMP traffic through the IPsec tunnel between the TOE and test workstation (encrypted). Verify traffic matching the ACL is encrypted and non-matching traffic, not disallowed by any other ACL, flows in plaintext. Create an ACL to deny SSH between the TOE and test workstation (dropped). Verify traffic matching the ACL is denied/dropped and non-matching traffic, not disallowed by any other ACL, flows in plaintext.
Findings: PASS

b. Test 2: The evaluator shall devise several tests that cover a variety of scenarios for packet processing. As with Test 1, the evaluator ensures both positive and negative test cases are constructed. These scenarios must exercise the range of possibilities for SPD entries and processing modes as outlined in the TSS and guidance documentation. Potential areas to cover include rules with overlapping ranges and conflicting entries, inbound and outbound packets, and packets that establish SAs as well as packets that belong to established SAs. The evaluator shall verify, via the audit trail and packet captures, for each scenario that the expected behavior is exhibited, and is consistent with both the TSS and the guidance documentation.

High-Level Test Description
According to the guidance documentation, ACL rules are processed sequentially and deny all undescribed traffic by default. When an ACL is applied to an ipsec-isakmp Crypto Map, traffic permitted within the ACL will flow through the configured IPSec tunnel. The evaluator took this and the Test 1 ACLs into consideration and devised the following ACL for this test.
ip access-list extended 113 10 permit ip host 10.20.10.3 any 20 deny tcp host 10.20.10.3 any eq 22 30 permit tcp host 10.20.10.101 eq 10101 10.20.10.0 0.0.0.255 eq 22 40 deny ip host 10.20.10.101 any 50 permit ip 10.100.1.0 0.0.0.255 any
Using the above ACL, verify packets are processed as described in the TSS.
Findings: PASS

FCS_IPSEC_EXT.1.2

364 The assurance activity for this element is performed in conjunction with the activities for FCS_IPSEC_EXT.1.1.

365 The evaluator uses the guidance documentation to configure the TOE to carry out the following tests:

366 The evaluator shall configure the SPD such that there is a rule for dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext. The evaluator may use the SPD that was created for verification of FCS_IPSEC_EXT.1.1. The evaluator shall construct a network packet that matches the rule to allow the packet to flow in plaintext and send that packet. The evaluator should observe that the network packet is passed to the proper destination interface with no modification. The evaluator shall then modify a field in the packet header; such that it no longer matches the evaluator-created entries (there may be a "TOE created" final entry that discards packets that do not match any previous entries). The evaluator sends the packet, and observes that the packet was dropped.

High-Level Test Description
Note that plaintext packet flow has been demonstrated in the FCS_IPSEC_EXT.1. Verify packets not matching any statement in the following ACL are denied by default.
ip access-list extended 113 10 permit ip host 10.20.10.3 any 20 deny tcp host 10.20.10.3 any eq 22 30 permit tcp host 10.20.10.101 eq 10101 10.20.10.0 0.0.0.255 eq 22 40 deny ip host 10.20.10.101 any 50 permit ip 10.100.1.0 0.0.0.255 any
Findings: PASS

FCS_IPSEC_EXT.1.3

367 The evaluator shall perform the following test(s) based on the selections chosen:

a. Test 1: If tunnel mode is selected, the evaluator uses the guidance documentation to configure the TOE to operate in tunnel mode and also configures a VPN peer to operate in tunnel mode. The evaluator configures the TOE and the VPN peer to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a connection from the TOE to connect to the VPN peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the tunnel mode.

High-Level Test Description
Initiate an IPsec connection with the WLC using tunnel mode and verify the connection is successful by inspection of the audit logs, traffic and console output.
Findings: PASS

b. Test 2: If transport mode is selected, the evaluator uses the guidance documentation to configure the TOE to operate in transport mode and also

configures a VPN peer to operate in transport mode. The evaluator configures the TOE and the VPN peer to use any of the allowed cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator then initiates a connection from the TOE to connect to the VPN peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the transport mode.

High-Level Test Description
Initiate an IPsec connection with the WLC using transport mode and verify the connection is successful by inspection of the audit logs, traffic and console output.
Findings: PASS

FCS_IPSEC_EXT.1.4

368 The evaluator shall configure the TOE as indicated in the guidance documentation configuring the TOE to use each of the supported algorithms, attempt to establish a connection using ESP, and verify that the attempt succeeds.

High-Level Test Description
Attempt to establish an IPsec connection using each of the supported ESP/Phase 2 encryption algorithms. Verify the connection is successful and the correct phase 2 encryption algorithm is used by inspection of the audit logs and associated traffic capture.
Findings: PASS

FCS_IPSEC_EXT.1.5

369 Tests are performed in conjunction with the other IPsec evaluation activities.

a) Test 1: If IKEv1 is selected, the evaluator shall configure the TOE as indicated in the guidance documentation and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator should then show that main mode exchanges are supported.

Test Not Applicable:	The TOE does not claim support for IKEv1.

b) Test 2: If NAT traversal is selected within the IKEv2 selection, the evaluator shall configure the TOE so that it will perform NAT traversal processing as described in the TSS and RFC 5996, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.

Test Not Applicable:	The TOE does not claim support for NAT traversal.

FCS_IPSEC_EXT.1.6

370 The evaluator shall configure the TOE to use the ciphersuite under test to encrypt the IKEv1 and/or IKEv2 payload and establish a connection with a peer device, which is configured to only accept the payload encrypted using the indicated ciphersuite. The evaluator will confirm the algorithm was that used in the negotiation.

High-Level Test Description
Attempt to establish an IPsec connection between the test workstation and the WLC using each supported Phase 1 encryption algorithm. Verify the connection is successful using the expected encryption algorithm by inspection of the audit logs and associated traffic capture.
Findings: PASS

FCS_IPSEC_EXT.1.7

[Updated per TD 0633]

371 When testing this functionality, the evaluator needs to ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered."

372 Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

a. Test 1: If 'number of bytes' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime in terms of the number of bytes allowed following the guidance documentation. The evaluator shall configure a test peer with a byte lifetime that exceeds the lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, and determine that once the allowed number of bytes through this SA is exceeded, a new SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 1 negotiation.

Test Not Applicable:	The TOE does not claim support for phase-1 SA lifetimes based on based on volume.

b. Test 2: If 'length of time' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime no later than 24 hours for the Phase 1 SA following the guidance documentation. The evaluator shall configure a test peer with a Phase 1 SA lifetime that exceeds the Phase 1 SA lifetime on the TOE.

High-Level Test Description
Configure the IPsec Phase 1 lifetime to be 24 hours and verify the WLC initiates a Phase 1 rekey prior to the expiration of the Phase 1 lifetime.
Findings: PASS

FCS_IPSEC_EXT.1.8

[Updated per TD 0633]

373 When testing this functionality, the evaluator needs to ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is

responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered."

374 Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

a. Test 1: If 'number of bytes' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime in terms of the number of bytes allowed following the guidance documentation. The evaluator shall configure a test peer with a byte lifetime that exceeds the lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, and determine that once the allowed number of bytes through this SA is exceeded, a new SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 2 negotiation.

High-Level Test Description
Configure the IPsec Phase 2 SA lifetime to be 4294967295 kB and verify it has been configured properly. Configure the IPsec Phase 2 SA lifetime to be 2560 kB to facilitate this test and verify the WLC initiates a Phase 2 SA renegotiation occurs prior to the expiration of the Phase 2 lifetime.
Findings: PASS

b. Test 2: If 'length of time' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime no later than 8 hours for the Phase 2 SA following the guidance documentation. The evaluator shall configure a test peer with a Phase 2 SA lifetime that exceeds the Phase 2 SA lifetime on the TOE.

High-Level Test Description
Configure the IPsec Phase 2 SA lifetime to be 8 hours and verify the WLC initiates a Phase 2 SA renegotiation occurs prior to the expiration of the Phase 2 lifetime.
Findings: PASS

FCS_IPSEC_EXT.1.10

375 Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

a) Test 1: If the first selection is chosen, the evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of the nonces meet the stipulations in the requirement.

Findings:	This selection is not made in the [ST].

b) Test 2: If the second selection is chosen, the evaluator shall check to ensure that, for each PRF hash supported, the TSS describes the process for generating each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of the nonces meet the stipulations in the requirement.

Findings:	The FCS_IPSEC_EXT.1 section of the TSS states the following, "The TOE supports Diffie-Hellman Groups 19 and 20. The length of the nonce is equal to that of the hash PRF used in the session establishment (for SHA-256 hash based PRF the nonce is 256-bits and for SHA-384 Hash based PRF the nonce is 384-bits) The TSF generates the secret value 'x' used in the IKEv2 Diffie-Hellman key exchange ('x' in gx mod p) using the NIST approved DRBG specified in FCS_RBG_EXT.1 and having possible lengths of 256 or 384 bits. When a random number is needed for a nonce, the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2128. The nonce is likewise generated using the AES-CTR DRBG" This satisfies the requirements in this PP and the length of the nonces meet the requirements of the ST of 128 bits and at least half the output size of the PRF hash (256 or 384 bits).

FCS_IPSEC_EXT.1.11

376 For each supported DH group, the evaluator shall test to ensure that all supported IKE protocols can be successfully completed using that particular DH group.

High-Level Test Description
Initiate an IPsec connection from the WLC to the test workstation using each claimed DH group. Verify the IPsec connection succeeds and the appropriate DH group is used by inspection of the audit logs and associated traffic capture.
Findings: PASS

FCS_IPSEC_EXT.1.12

377 The evaluator simply follows the guidance to configure the TOE to perform the following tests.

a. Test 1: This test shall be performed for each version of IKE supported. The evaluator shall successfully negotiate an IPsec connection using each of the supported algorithms and hash functions identified in the requirements.

Findings:	This test was done as part of FCS_IPSEC_EXT.1.4.

b. Test 2: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an SA for ESP that selects an encryption algorithm with more strength than that being used for the IKE SA (i.e., symmetric algorithm with a key size larger than that being used for the IKE SA). Such attempts should fail.

Findings:	The [AGD] instructs the administrator to explicitly ensure the encryption algorithm used for the IKE SA is greater than or equal to the strength of the encryption algorithm used for the ESP SA. Thus, it is the administrator's responsibility to ensure this requirement is satisfied.

c. Test 3: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an IKE SA using an algorithm that is not one of the supported algorithms and hash functions identified in the requirements. Such an attempt should fail.

High-Level Test Description
Configure the IPsec peer to use an unsupported algorithm within the IKE SA proposal and initiate an IPsec connection from the TOE to the test workstation. Verify the connection fails by inspection of the audit log and associated traffic capture.
Findings: PASS

d. Test 4: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an SA for ESP (assumes the proper parameters where used to establish the IKE SA) that selects an encryption algorithm that is not identified in FCS_IPSEC_EXT.1.4. Such an attempt should fail.

High-Level Test Description
Configure the IPsec peer to use an unsupported algorithm within the ESP SA proposal and initiate an IPsec connection from the TOE to the test workstation. Verify the connection fails by inspection of the audit log and associated traffic capture.
Findings: PASS

FCS_IPSEC_EXT.1.13

378 For efficiency sake, the testing that is performed may be combined with the testing for FIA_X509_EXT.1, FIA_X509_EXT.2 (for IPsec connections), and FCS_IPSEC_EXT.1.1.

FCS_IPSEC_EXT.1.14

379 In the context of the tests below, a valid certificate is a certificate that passes FIA_X509_EXT.1 validation checks but does not necessarily contain an authorized subject.

380 The evaluator shall perform the following tests:

381 Test 1: [conditional] For each CN/identifier type combination selected, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the field in the peer's presented certificate and shall verify that the IKE authentication succeeds. If the TOE prioritizes CN checking over SAN (through explicit configuration of the field when specifying the reference identifier or prioritization rules), the evaluator shall also configure the SAN so it contains an incorrect identifier of the correct type (e.g. the reference identifier on the TOE is example.com, the CN=example.com, and the SAN:FQDN=otherdomain.com) and verify that IKE authentication succeeds.

Findings:	There are no CN/Identifier types selected in the ST.

382 Test 2: [conditional] For each SAN/identifier type combination selected, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the field in the peer's presented certificate and shall verify that the IKE authentication succeeds. If the TOE prioritizes SAN checking over CN (through explicit specification of the field when specifying the reference identifier or prioritization rules), the evaluator shall also configure the CN so it contains an incorrect identifier formatted to be the same type (e.g. the reference identifier on the TOE is DNS-ID; identify certificate has an identifier in SAN with correct DNS-ID, CN with incorrect DNS-ID (and not a different type of identifier)) and verify that IKE authentication succeeds.

High-Level Test Description
Initiate an IPsec connection from the test workstation to the WLC using a peer certificate with the expected identifier. Verify the connection succeeds by inspection of the audit logs and associated traffic capture.
Findings: PASS

383 Test 3: [conditional] For each CN/identifier type combination selected, the evaluator shall:

a. Create a valid certificate with the CN so it contains the valid identifier followed by '\0'. If the TOE prioritizes CN checking over SAN (through explicit specification of the field when specifying the reference identifier or prioritization rules) for the same identifier type, the evaluator shall configure the SAN so it matches the reference identifier.

Findings:	There are no CN/Identifier types selected in the ST.

b. Configure the peer's reference identifier on the TOE (per the administrative guidance) to match the CN without the '\0' and verify that IKE authentication fails.

Findings:	There are no CN/Identifier types selected in the ST.

384 Test 4: [conditional] For each SAN/identifier type combination selected, the evaluator shall:

a) Create a valid certificate with an incorrect identifier in the SAN. The evaluator shall configure a string representation of the correct identifier in the DN. If the TOE prioritizes CN checking over SAN (through explicit specification of the field when specifying the reference identifier or prioritization rules) for the same identifier type, the addition/modification shall be to any non-CN field of the DN. Otherwise, the addition/modification shall be to the CN.

Findings:	This test is covered below in Test 4 b).

b) Configure the peer's reference identifier on the TOE (per the administrative guidance) to match the correct identifier (expected in the SAN) and verify that IKE authentication fails.

High-Level Test Description
Initiate an IPsec connection from the test workstation to the WLC using a peer certificate that has a matching reference identifier in the CN and a non-matching reference identifier in the SAN. Verify the IPsec connection fails by inspection of the audit logs and associated traffic capture. Repeat for each SAN/identifier type combination selected in the requirement.
Findings: PASS

385 Test 5: [conditional] If the TOE supports DN identifier types, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the subject DN in the peer's presented certificate and shall verify that the IKE authentication succeeds.

Findings:	There are no DN/identifier types selected in the ST.

386 Test 6: [conditional] If the TOE supports DN identifier types, to demonstrate a bit-wise comparison of the DN, the evaluator shall create the following valid certificates and verify that the IKE authentication fails when each certificate is presented to the TOE:

a. Duplicate the CN field, so the otherwise authorized DN contains two identical CNs.

Findings:	There are no DN/identifier types selected in the ST.

b. Append '\0' to a non-CN field of an otherwise authorized DN.

Findings:	There are no DN/identifier types selected in the ST.

4.2.5 FCS_SSHS_EXT.1 SSH Server

4.2.5.1 TSS

FCS_SSHS_EXT.1.2

[Updated per TD 0631]

387 The evaluator shall check to ensure that the TSS contains a list of supported public key algorithms that are accepted for client authentication and that this list is consistent with signature verification algorithms selected in FCS_COP.1/SigGen (e.g., accepting EC keys requires corresponding Elliptic Curve Digital Signature algorithm claims).

388 The evaluator shall confirm that the TSS includes the description of how the TOE establishes a user identity when an SSH client presents a public key or X.509v3

certificate. For example, the TOE could verify that the SSH client's presented public key matches one that is stored within the SSH server's authorized_keys file.

389 If password-based authentication method has been selected in the FCS_SSHS_EXT.1.2, then the evaluator shall confirm its role in the authentication process is described in the TSS.

Findings:	[ST] / TOE Summary Specification states, "The TSF's SSH transport implementation supports the following public-key algorithms for both Client Authentication and Hostkey authentication: - rsa-sha2-256 - rsa-sha2-512" [ST] / TOE Summary Specification – "When the SSH client presents a public key, the TSF verifies it matches with a configured Administrator account. If the presented public key does not match with a configured Administrator account, access is denied." The evaluator confirmed the role of password-based authentication process is described in the TSS and the supported public key algorithms accepted for client authentication is consistent with the signature verification algorithms in FCS_COP.1/SigGen.

FCS_SSHS_EXT.1.3

390 The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled.

Findings:	[ST] / TOE Summary Specification states "SSHv2 connections will be dropped if the TOE receives a packet larger than 65,535 bytes. Large packets are detected by the SSHv2 implementation and dropped internal to the SSH process."

FCS_SSHS_EXT.1.4

391 The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.

Findings:	[ST] /TOE Summary Specification states, "The TSF's SSH transport implementation supports the following encryption algorithms: ■ aes128-cbc ■ aes128-ctr ■ aes256-cbc ■ aes256-ctr "All connection attempts from remote SSH clients requesting any other encryption algorithm is denied." The evaluator confirmed these encryption algorithms are identical to those listed for this component.

FCS_SSHS_EXT.1.5

[Updated per TD 0631]

392 The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the SSH server's host public key algorithms supported are specified and that they are identical to those listed for this component.

Findings:	[ST] /TOE Summary Specification states, **“**“The TSF’s SSH transport implementation supports the following public-key algorithms for both Client Authentication and Hostkey authentication: - rsa-sha2-256 - rsa-sha2-512“”. The evaluator confirmed this is identical to those listed for this component.

FCS_SSHS_EXT.1.6

393 The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component.

Findings:	[ST] / TOE Summary Specification lists, “The TSF’s SSH transport implementation supports the following MAC algorithms: ■ hmac-sha2-256 ■ hmac-sha2-512” The evaluator confirmed that this list corresponds to the list in this component.

FCS_SSHS_EXT.1.7

394 The evaluator shall check the TSS to ensure that it lists the supported key exchange algorithms, and that that list corresponds to the list in this component.

Findings:	[ST] / TOE Summary Specification states, “The TSF’s SSH key exchange implementation supports ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521.” The evaluator confirmed that this corresponds to the component.

FCS_SSHS_EXT.1.8

395 The evaluator shall check that the TSS specifies the following:

a) Both thresholds are checked by the TOE.

b) Rekeying is performed upon reaching the threshold that is hit first.

Findings:	[ST] / TOE Summary Specification states, “The TSF's SSH implementation will perform a rekey after no longer than one hour or more than one gigabyte of data has been transmitted with the same session key. Both thresholds are checked. Rekeying is performed upon reaching whichever threshold is met first.”

4.2.5.2 Guidance Documentation

FCS_SSHS_EXT.1.4

396 The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

Findings:	The SSH subsection of the section, Preparative Procedures and Operational Guidance for the TOE provides instructions on configuring the TOE SSH server

functionality. The instructions to configure encryption algorithms correspond to the description given in the TSS.

The subsection states,

“7. Specify the allowed encryption algorithms and the order they are to be supported

WLC(config)# ip ssh server algorithm encryption aes256-cbc aes256-ctr aes128-cbc aes128-ctr”

FCS_SSHS_EXT.1.5

397 The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

Findings: The SSH subsection of the section, Preparative Procedures and Operational Guidance for the TOE provides instructions on configuring the TOE SSH server functionality. The instructions to configure public key algorithms correspond to the description given in the TSS.

The subsection states,

“a. Configure Host Key Algorithms for SSH public-key based authentication

WLC(config)# ip ssh server algorithm hostkey rsa-sha2-256 rsa-sha2-512”

FCS_SSHS_EXT.1.6

398 The evaluator shall also check the guidance documentation to ensure that it contains instructions to the Security Administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the “none” MAC algorithm is not allowed).

Findings: The SSH subsection of the section, Preparative Procedures and Operational Guidance for the TOE provides instructions on configuring the TOE SSH server functionality. The instructions provided to configure MAC algorithms correspond to the description given in the TSS.

The subsection states,

“8. Specify the allowed Message Authentication Code (MAC) algorithms and the order they are to be supported

WLC(config)# ip ssh server algorithm mac hmac-sha2-512 hmac-sha2-256”

FCS_SSHS_EXT.1.7

399 The evaluator shall also check the guidance documentation to ensure that it contains instructions to the Security Administrator on how to ensure that only the allowed key exchange algorithms are used in SSH connections with the TOE.

Findings: The SSH subsection of the section, Preparative Procedures and Operational Guidance for the TOE provides instructions on configuring the TOE SSH server functionality including allowed key exchange algorithms.

The subsection states,

“6. Configure the SSH Server Key Exchange

	WLC(config)# ip ssh server algorithm kex ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521”

FCS_SSHS_EXT.1.8

400 If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, then the evaluator shall check that the guidance documentation describes how to configure those thresholds. Either the allowed values are specified in the guidance documentation and must not exceed the limits specified in the SFR (one hour of session time, one gigabyte of transmitted traffic) or the TOE must not accept values beyond the limits specified in the SFR. The evaluator shall check that the guidance documentation describes that the TOE reacts to the first threshold reached.

Findings:	The SSH subsection of the section, Preparative Procedures and Operational Guidance for the TOE provides instructions on configuring the TOE SSH server functionality including rekey thresholds. The threshold configuration instructions correspond to the limits specified in the SFR. The subsection states, “11. SSH connections with the same session keys cannot be used longer than one hour, and with no more than one gigabyte of transmitted data. In the steps below configure a time-based and volume-based (in kilobytes) rekey values. Note: Values can be configured to be lower if desired. The minimum time value is 10 minutes. The minimum volume value is 100 kilobytes. WLC(config)# ip ssh rekey time 60 WLC(config)# ip ssh rekey volume 1000000”

4.2.5.3 Tests

FCS_SSHS_EXT.1.2

[Updated per TD 0631]

401 Test 1: For each supported client public-key authentication algorithm, the evaluator shall configure a remote client to present a public key corresponding to that authentication method (e.g., 2048-bit RSA key when using ssh-rsa public key). The evaluator shall establish sufficient separate SSH connections with an appropriately configured remote non-TOE SSH client to demonstrate the use of all applicable public key algorithms. It is sufficient to observe the successful completion of the SSH Authentication Protocol to satisfy the intent of this test.

Findings:	This test is done in FCS_SSHS_EXT.1.5.

402 Test 2: The evaluator shall choose one client public key authentication algorithm supported by the TOE. The evaluator shall generate a new client key pair for that supported algorithm without configuring the TOE to recognize the associated public key for authentication. The evaluator shall use an SSH client to attempt to connect to the TOE with the new key pair and demonstrate that authentication fails.

High-Level Test Description
Generate two new client key pairs for a supported public key authentication algorithm supported by the TOE (i.e. ssh-rsa).

High-Level Test Description
Configure the TOE to recognize one of the newly created public keys for SSH authentication. Attempt to login to the TOE using the unrecognized public key. Verify the SSH authentication fails and that an appropriate audit message is emitted.
Findings: PASS

403 Test 3: [Conditional] If password-based authentication method has been selected in the FCS_SSHS_EXT.1.2, the evaluator shall configure the TOE to accept password-based authentication and demonstrate that user authentication succeeds when the correct password is provided by the connecting SSH client.

Findings:	This test is covered by FIA_UIA_EXT.1 and FIA_UAU_EXT.2.

404 Test 4: [Conditional] If password-based authentication method has been selected in the FCS_SSHS_EXT.1.2, the evaluator shall configure the TOE to accept password-based authentication and demonstrate that user authentication fails when the incorrect password is provided by the connecting SSH client.

Findings:	This test is covered by FIA_UIA_EXT.1 and FIA_UAU_EXT.2.

FCS_SSHS_EXT.1.3

405 The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

High-Level Test Description
Transmit a SSH packet larger than the expected TOE buffer size (32768 bytes) and show that the TOE rejects the packet in some way.
Findings: PASS

FCS_SSHS_EXT.1.4

406 The evaluator must ensure that only claimed ciphers and cryptographic primitives are used to establish an SSH connection. To verify this, the evaluator shall start session establishment for an SSH connection from a remote client (referred to as 'remote endpoint' below). The evaluator shall capture the traffic exchanged between the TOE and the remote endpoint during protocol negotiation (e.g. using a packet capture tool or information provided by the endpoint, respectively). The evaluator shall verify from the captured traffic that the TOE offers all the ciphers defined in the TSS for the TOE for SSH sessions, but no additional ones compared to the definition in the TSS. The evaluator shall perform one successful negotiation of an SSH session to verify that the TOE behaves as expected. It is sufficient to observe the successful negotiation of the session to satisfy the intent of the test. If the evaluator detects that not all ciphers defined in the TSS for SSH are supported by the TOE and/or the TOE supports one or more additional ciphers not defined in the TSS for SSH, the test shall be regarded as failed.

High-Level Test Description
Using an SSH client, connect to the TOE server and capture the TOE server's advertised supported cipher algorithms. Verify that the advertised set matches the claimed set. Forcibly use an SSH client to connect using only one of those ciphers and show that the connection is successful.
Findings: PASS

FCS_SSHS_EXT.1.5

[Updated per TD 0631]

407 Test objective: This test case is meant to validate that the TOE server will support host public keys of the claimed algorithm types.

408 Test 1: The evaluator shall configure (only if required by the TOE) the TOE to use each of the claimed host public key algorithms. The evaluator will then use an SSH client to confirm that the client can authenticate the TOE server public key using the claimed algorithm. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

High-Level Test Description
Using an SSH client, connect to the TOE server using the specified public key algorithms in turn. This requires the TOE to be loaded with a public key corresponding to the key pair.
Findings: PASS

409 Test objective: This negative test case is meant to validate that the TOE server does not support host public key algorithms that are not claimed.

410 Test 2: The evaluator shall configure a non-TOE SSH client to only allow it to authenticate an SSH server host public key algorithm that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection from the non-TOE SSH client to the TOE SSH server and observe that the connection is rejected.

High-Level Test Description
Using an SSH client configured to only allow ssh-rsa host key algorithms, attempt to establish an SSH session with the TOE. Verify the SSH connection fails.
Findings: PASS

FCS_SSHS_EXT.1.6

411 Test 1: [conditional, if an HMAC or AEAD_AES_*_GCM algorithm is selected in the ST] The evaluator shall establish an SSH connection using each of the algorithms, except "implicit", specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

412 Note: To ensure the observed algorithm is used, the evaluator shall ensure a non-aes*-gcm@openssh.com encryption algorithm is negotiated while performing this test.

High-Level Test Description
Using an SSH client, forcibly negotiate only the claimed integrity algorithms and show that they are accepted to form a successful connection.

High-Level Test Description
Findings: PASS

413 Test 2: [conditional, if an HMAC or AEAD_AES_*_GCM algorithm is selected in the ST] The evaluator shall configure an SSH client to only allow a MAC algorithm that is not included in the ST selection. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.

414 Note: To ensure the proposed MAC algorithm is used, the evaluator shall ensure a non-aes*-gcm@openssh.com encryption algorithm is negotiated while performing this test.

High-Level Test Description
Using an SSH client, forcibly negotiate an integrity algorithm which is not claimed by the TOE and show that it results in a failed connection.
Findings: PASS

FCS_SSHS_EXT.1.7

415 Test 1: The evaluator shall configure an SSH client to only allow the diffie-hellman-group1-sha1 key exchange. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.

High-Level Test Description
Using an SSH client, forcibly negotiate the diffie-hellman-group1-sha1 key exchange algorithm which is not supported by the TOE and show that it results in a failed connection.
Findings: PASS

416 Test 2: For each allowed key exchange method, the evaluator shall configure an SSH client to only allow that method for key exchange, attempt to connect from the client to the TOE, and observe that the attempt succeeds.

High-Level Test Description
Using an SSH client, forcibly negotiate each of the claimed key exchange algorithms in turn and show that it results in a successful connection.
Findings: PASS

FCS_SSHS_EXT.1.8

417 The evaluator needs to perform testing that rekeying is performed according to the description in the TSS. The evaluator shall test both, the time-based threshold and the traffic-based threshold.

418 For testing of the time-based threshold, the evaluator shall use an SSH client to connect to the TOE and keep the session open until the threshold is reached. The evaluator shall verify that the SSH session has been active longer than the threshold value and shall verify that the TOE initiated a rekey (the method of verification shall be reported by the evaluator).

419 Testing does not necessarily have to be performed with the threshold configured at the maximum allowed value of one hour of session time but the value used for testing shall not exceed one hour. The evaluator needs to ensure that the rekeying has been initiated by the TOE and not by the SSH client that is connected to the TOE.

High-Level Test Description
Set the SSH time-based rekey threshold to 10 minutes. Using a custom SSH client, connect to the TOE and trickle data over the channel to avoid disconnection due to idle timeout. Ensure that the TOE rekeys before the rekey threshold. Ensure that the TOE is responsible for sending the rekey initiation.
Findings: PASS

420 For testing of the traffic-based threshold the evaluator shall use the TOE to connect to an SSH client and shall transmit data to and/or receive data from the TOE within the active SSH session until the threshold for data protected by either encryption key is reached. It is acceptable if the rekey occurs before the threshold is reached (e.g. because the traffic is counted according to one of the alternatives given in the Application Note for FCS_SSHS_EXT.1.8).

421 The evaluator shall verify that more data has been transmitted within the SSH session than the threshold allows and shall verify that the TOE initiated a rekey (the method of verification shall be reported by the evaluator).

422 Testing does not necessarily have to be performed with the threshold configured at the maximum allowed value of one gigabyte of transferred traffic, but the value used for testing shall not exceed one gigabyte. The evaluator needs to ensure that the rekeying has been initiated by the TOE and not by the SSH client that is connected to the TOE.

High-Level Test Description
Set the SSH volume-based rekey threshold to 100kB. Copy a file with a size exceeding the SSH volume-based rekey threshold to/from the TOE via SSH. Verify the TOE initiates a SSH rekey once the 100kB threshold is reached.
Findings: PASS

423 If one or more thresholds that are checked by the TOE to fulfil the SFR are configurable, the evaluator needs to verify that the threshold(s) can be configured as described in the guidance documentation and the evaluator needs to test that modification of the thresholds is restricted to Security Administrators (as required by FMT_MOF.1/Functions).

High-Level Test Description
Log into the TOE as a Security Administrator and modify the time-based and volume-based rekeying thresholds as described in the guidance documentation (60 minutes, 1GB). Verify an audit message is emitted by the TOE indicating the threshold has been changed. Verify the new limit takes effect by rerunning the time-based or volume-based test case above with the new limit. Log into the TOE as an unprivileged user and verify attempts to change the SSH rekey thresholds fails.
Findings: PASS

424 In cases where data transfer threshold could not be reached due to hardware limitations it is acceptable to omit testing of this (SSH rekeying based on data transfer threshold) threshold if both the following conditions are met:

a. An argument is present in the TSS section describing this hardware-based limitation and

b. All hardware components that are the basis of such argument are definitively identified in the ST. For example, if specific Ethernet Controller or WiFi radio chip is the root cause of such limitation, these chips must be identified.

Findings:	The TOE does not have hardware limitations.

4.2.6 FCS_TLSC_EXT.1/RADsec Extended: TLS Client Protocol without mutual authentication

4.2.6.1 TSS

FCS_TLSC_EXT.1.1

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

Findings:	[ST] / TOE Summary Specification species the following ciphersuite: ■ TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268 This is consistent with those listed in the component.

FCS_TLSC_EXT.1.2

426 The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the administrator/application-configured reference identifier, including which types of reference identifiers are supported (e.g. application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported.

Findings:	[ST] / TOE Summary Specification states, "When establishing a TLS connection, the TOE supports reference identifiers of type DNS-ID and IP address and will seek a match to the DNS domain name or IP address respectively in the subjectAltName extension." "The TOE does not support the use of wildcards within certificates and does not support certificate pinning."

427 Note that where a TLS channel is being used between components of a distributed TOE for FPT_ITT.1, the requirements to have the reference identifier established by the user are relaxed and the identifier may also be established through a "Gatekeeper" discovery process. The TSS should describe the discovery process and highlight how the reference identifier is supplied to the "joining" component. Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1 and the ST author selected attributes from RFC 5280, the evaluator shall ensure the TSS describes which attribute type, or combination of attributes types, are used by the client to match the presented identifier with the configured identifier. The evaluator shall ensure the TSS presents an argument how the attribute type, or

combination of attribute types, uniquely identify the remote TOE component; and the evaluator shall verify the attribute type, or combination of attribute types, is sufficient to support unique identification of the maximum supported number of TOE components.

Findings:	FCS_TLSC_EXT.1 is not used for FPT_ITT.1.

428 If IP addresses are supported in the CN as reference identifiers, the evaluator shall ensure that the TSS describes the TOE's conversion of the text representation of the IP address in the CN to a binary representation of the IP address in network byte order. The evaluator shall also ensure that the TSS describes whether canonical format (RFC 5952 for IPv6, RFC 3986 for IPv4) is enforced.

Findings:	The ST does not claim support for IP addresses in the CN.

FCS_TLSC_EXT.1.4

429 The evaluator shall verify that TSS describes the Supported Elliptic Curves Extension and whether the required behaviour is performed by default or may be configured.

Findings:	The ST does not claim support for the Supported Elliptic Curves Extension.

4.2.6.2 Guidance Documentation

FCS_TLSC_EXT.1.1

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

Findings:	The TLS-RADsec subsection of the section, Preparative Procedures and Operational Guidance for the TOE in the [AGD] provides instructions on configuring the TOEs RADIUS over TLS client functionality. The instructions provided correspond to the description given in the TSS.

FCS_TLSC_EXT.1.2

431 The evaluator shall ensure that the operational guidance describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE includes support for IP addresses, the evaluator shall ensure that the operational guidance provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.

Findings:	The TLS-RADsec subsection of the section, Preparative Procedures and Operational Guidance for the TOE in the [AGD] indicates the SAN extension is supported and provides instructions on configuring peer reference identifiers. IP addresses are supported identifiers and an appropriate warning/policy recommendation for secure TOE use is provided in the TLS-RADsec subsection. The subsection states, And

	"The TOE may be configured to perform identity verification using either an IP address or DNS Name in the SAN extension of the X.509 certificate. This is covered in step 14 in the section below. The Administrator is advised to follow the security policies and procedures of their organization if using an IP address to verify RADsec server identity." "14. Specify the Reference Identifier for the Peer using DNS name or IP address. WLC(config-radius-server)# tls match-server-identity hostname <DNS Name> WLC(config-radius-server)# tls match-server-identity ip-address <IP Address>"

432 Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1, the SFR selects attributes from RFC 5280, and FCO_CPC_EXT.1.2 selects "no channel"; the evaluator shall verify the guidance provides instructions for establishing unique reference identifiers based on RFC5280 attributes.

Findings:	The "no channel" selection is not made in the FCO_CPC_EXT.1.2 SFR of the [ST]. RADIUS over TLS is not used between TOE components.

FCS_TLSC_EXT.1.4

433 If the TSS indicates that the Supported Elliptic Curves/Supported Groups Extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the Supported Elliptic Curves/Supported Groups Extension.

Findings:	The TSS does not describe any configuration requirement for the Supported Groups Extension as it is not claimed by the TOE for RADsec purposes.

4.2.6.3 Tests

FCS_TLSC_EXT.1.1

434 Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites 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 HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to negotiate all claimed ciphersuites.
Findings: PASS

435 Test 2: The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and verify that a connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field, and a connection is not established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.

High-Level Test Description
Construct two X.509 certificates, one with an extendedKeyUsage with ‘serverAuth’ and another without. Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server and show that the handshake using the X.509 certificate with appropriate EKU succeeds and the handshake using the X.509 certificate without the EKU fails.
Findings: PASS

436 Test 3: The evaluator shall send a server certificate in the TLS connection that does not match the server-selected ciphersuite (for example, send an ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite). The evaluator shall verify that the TOE disconnects after receiving the server’s Certificate handshake message.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server using any of the claimed ciphersuites. The Lightship TLS server will send back an otherwise validly constructed server certificate which does not match the requested the ciphersuite.
Findings: PASS

437 Test 4: The evaluator shall perform the following 'negative tests':

a) The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server using the TLS_NULL_WITH_NULL_NULL (cipher ID 0x0000).
Findings: PASS

b) Modify the server’s selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server sending a non-negotiated ciphersuite.
Findings: PASS

c) [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension the evaluator shall configure the server to perform an ECDHE or DHE key exchange in the TLS connection using a non-supported curve/group (for example P-192) and shall verify that the TOE disconnects after receiving the server’s Key Exchange handshake message.

Findings:	The TOE does not present Supported Elliptic Curves/Supported Groups Extension.

438 Test 5: The evaluator performs the following modifications to the traffic:

a. Change the TLS version selected by the server in the Server Hello to a non-supported TLS version and verify that the client rejects the connection.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server advertising an incorrect TLS version.
Findings: PASS

b. [conditional]: If using DHE or ECDH, modify the signature block in the Server's Key Exchange handshake message, and verify that the handshake does not finished successfully, 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.

Findings:	The TOE does not claim support for DHE or ECDH key exchanges.

439 Test 6: The evaluator performs the following 'scrambled message tests':

a. Modify a byte in the Server Finished handshake message and verify that the handshake does not finish successfully and no application data flows.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server sending a mangled finished message.
Findings: PASS

b. Send a garbled message from the server after the server has issued the ChangeCipherSpec message and verify that the handshake does not finish successfully and no application data flows.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with the test server that sends a mangled finished message after issuing the ChangeCipherSpec message.
Findings: PASS

c. Modify at least one byte in the server's nonce in the Server Hello handshake message and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server sending a modified nonce value. Do this once for a non-DHE ciphersuite and once for a DHE or ECDHE key exchange ciphersuite if applicable.
Findings: PASS

FCS_TLSC_EXT.1.2

440 Note that the following tests are marked conditional and are applicable under the following conditions:

a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.

or

b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable

or

c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.

441 Note that for some tests additional conditions apply.

442 IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:

- IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.
- IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.

443 The evaluator shall configure the reference identifier per the AGD guidance and perform the following tests during a TLS connection:

Note:	The TOE does not provide for, nor claim, any administrator-defined override mechanism for validating that the reference identifier matches that on the certificates for claimed TLS channels. Therefore, all of the following tests are applicable in the context of FCS_TLSC_EXT.1.

a. Test 1 [conditional]: 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. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the CN.

   Remark: 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.

High-Level Test Description
Force the TOE client to attempt a handshake with an OpenSSL s_server sub-application sending X.509 certificates that have the characteristics required by the test.
Findings: PASS

b. Test 2 [conditional]: 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 (e.g. IPv4, IPv6, FQDN, URI). When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the SAN.

High-Level Test Description
Force the TOE client to attempt a handshake with an OpenSSL s_server sub-application sending X.509 certificates that have the characteristics required by the test.
Findings: PASS

c. 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. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.

Findings:	The TOE mandates the presence of the SAN extension.

d. Test 4 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, SRV).

High-Level Test Description
Force the TOE client to attempt a handshake with an OpenSSL s_server sub-application sending X.509 certificates that have the characteristics required by the test.
Findings: PASS

e. Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID):

a) [conditional]: 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.

High-Level Test Description
Force the TOE client to attempt a handshake with an OpenSSL s_server sub-application sending X.509 certificates that have the characteristics required by the test.
Findings: PASS

b) [conditional]: 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 succeeds, if wildcards are supported, or fails if wildcards are not supported. 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.com) and verify that the connection fails. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)

High-Level Test Description
Force the TOE client to attempt a handshake with an OpenSSL s_server sub-application sending X.509 certificates that have the characteristics required by the test.
Findings: PASS

[Updated per TD 0634]

444 Objective: The objective of this test is to ensure the TOE is able to differentiate between IP address identifiers that are not allowed to contain wildcards and other types of identifiers that may contain wildcards.

f. Test 6: [conditional] If IP address identifiers supported in the SAN or CN, the evaluator shall present a server certificate that contains a CN that matches the reference identifier, except one of the groups has been replaced with a wildcard asterisk (*) (e.g. CN=*.168.0.1 when connecting to 192.168.0.1...

This negative test corresponds to the following section of the Application Note 64/105: "The exception being, the use of wildcards is not supported when using IP address as the reference identifier."

High-Level Test Description
Force the TOE client to attempt a handshake with an OpenSSL s_server sub-application sending X.509 certificates that have the characteristics required by the test.
Findings: PASS

g. Test 7 [conditional]: If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator modifies each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier):

1) The evaluator shall present a server certificate that does not contain an identifier in the Subject (DN) attribute type(s) that matches the reference identifier. The evaluator shall verify that the connection fails.

2) The evaluator shall present a server certificate that contains a valid identifier as an attribute type other than the expected attribute type (e.g. if the TOE is configured to expect id-at-serialNumber=correct_identifier, the certificate could instead include id-at-name=correct_identifier), and does not contain the SAN extension. The evaluator shall verify that the connection fails. Remark: 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 this test.

3) The evaluator shall present a server certificate that contains a Subject attribute type that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.

4) The evaluator shall confirm that all use of wildcards results in connection failure regardless of whether the wildcards are used in the left or right side of the presented identifier. (Remark: Use of wildcards is not addressed within RFC 5280.)

Findings:	The ST does not claim FPT_ITT.1 with RFC 5280.

FCS_TLSC_EXT.1.3

445 The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

446 Test 1: Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds, and a trusted channel can be established.

Findings:	This test is performed as part of FIA_X509_EXT.1.1/Rev (RadSec) Test 1.

447 Test 2: The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g. trusted channel was established) covering the types of failure for which an override mechanism is defined.

Findings:	This test case is performed as part of FIA_X509_EXT.1/Rev (RadSec). Appropriate override mechanisms are verified.

448 Test 3 [conditional]: The purpose of this test to verify that only selected certificate validation failures could be administratively overridden. If any override mechanism is defined for failed certificate validation, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative override available to accept such certificate.

Findings:	This test case is performed as part of FIA_X509_EXT.1/Rev (RadSec). Appropriate override mechanisms are verified.

FCS_TLSC_EXT.1.4

449 Test 1 [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension, the evaluator shall configure the server to perform ECDHE or DHE (as applicable) 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.

Findings:	The TOE does not support the Supported Elliptic Curves/Supported Groups Extension.

4.2.7 FCS_TLSC_EXT.1/EST Extended: TLS Client Protocol without mutual authentication

4.2.7.1 TSS

FCS_TLSC_EXT.1.1

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

Findings:	[ST] / TOE Summary Specification specifies the following ciphersuites: TLS communication between itself and an EST server supporting the following ciphersuites:

■	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
■	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
■	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
■	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
■	TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
■	TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
■	TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
■	TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
■	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
■	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
■	TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
■	TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
■	TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
■	TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
	This is consistent with those listed in the component.

FCS_TLSC_EXT.1.2

451 The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the administrator/application-configured reference identifier, including which types of reference identifiers are supported (e.g. application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported.

Findings:	[ST] / TOE Summary Specification states, "When establishing a TLS connection, the TOE supports reference identifiers of type DNS-ID and IP address and will seek a match to the DNS domain name or IP address respectively in the subjectAltName extension." "The TOE does not support the use of wildcards within certificates and does not support certificate pinning."

452 Note that where a TLS channel is being used between components of a distributed TOE for FPT_ITT.1, the requirements to have the reference identifier established by the user are relaxed and the identifier may also be established through a "Gatekeeper" discovery process. The TSS should describe the discovery process and highlight how the reference identifier is supplied to the "joining" component. Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1 and the ST author selected attributes from RFC 5280, the evaluator shall ensure the TSS describes which attribute type, or combination of attributes types, are used by the client to match the presented identifier with the configured identifier. The evaluator shall ensure the TSS presents an argument how the attribute type, or combination of attribute types, uniquely identify the remote TOE component; and the evaluator shall verify the attribute type, or combination of attribute types, is sufficient

to support unique identification of the maximum supported number of TOE components.

Findings:	FCS_TLSC_EXT.1 is not used for FPT_ITT.1.

453 If IP addresses are supported in the CN as reference identifiers, the evaluator shall ensure that the TSS describes the TOE's conversion of the text representation of the IP address in the CN to a binary representation of the IP address in network byte order. The evaluator shall also ensure that the TSS describes whether canonical format (RFC 5952 for IPv6, RFC 3986 for IPv4) is enforced.

Findings:	The ST does not claim support for IP addresses in the CN.

FCS_TLSC_EXT.1.4

454 The evaluator shall verify that TSS describes the Supported Elliptic Curves Extension and whether the required behaviour is performed by default or may be configured.

Findings:	[ST] / TOE Summary Specification states, "For TLS 1.2 connections to the EST server, the TSF presents secp256r1, secp384r1, and secp521r1 and no other curves in the Supported Group extension of the Client Hello. This behavior is implemented by default and is not configurable."

4.2.7.2 Guidance Documentation

FCS_TLSC_EXT.1.1

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

Findings:	The CC Mode subsection of the section, Preparative Procedures and Operational Guidance for the TOE in the [AGD] provides instructions on configuring the TOEs TLS client functionality for EST purposes. The instructions provided correspond to the description given in the TSS.

FCS_TLSC_EXT.1.2

456 The evaluator shall ensure that the operational guidance describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE includes support for IP addresses, the evaluator shall ensure that the operational guidance provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.

Findings:	The CC Mode subsection of the section, Preparative Procedures and Operational Guidance for the TOE in the [AGD] indicates the SAN extension is supported and provides instructions on configuring peer reference identifiers. IP addresses are supported identifiers and an appropriate warning/policy recommendation for secure TOE use is provided in the CC Mode subsection. The subsection states,

"3. Specify the SAN (alt-subject-name) field together with the matching criteria of equal and the value to match. In this example the value to match is estserver.cisco.com. WLC(ca-certificate-map)# alt-subject-name eq estserver.cisco.com"

457 Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1, the SFR selects attributes from RFC 5280, and FCO_CPC_EXT.1.2 selects "no channel"; the evaluator shall verify the guidance provides instructions for establishing unique reference identifiers based on RFC5280 attributes.

Findings:	The "no channel" selection is not made in the FCO_CPC_EXT.1.2 SFR of the [ST]. TLS for EST purposes is not used between TOE components.

FCS_TLSC_EXT.1.4

458 If the TSS indicates that the Supported Elliptic Curves/Supported Groups Extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the Supported Elliptic Curves/Supported Groups Extension.

Findings:	The TSS does not describe any configuration requirement for the Supported Groups Extension on the TOE for EST purposes.

4.2.7.3 Tests

FCS_TLSC_EXT.1.1

459 Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites 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 HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to negotiate all specifically claimed ciphersuites.
Findings: PASS

460 Test 2: The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and verify that a connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field, and a connection is not established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.

High-Level Test Description
Construct two X.509 certificates: one with an extendedKeyUsage with ‘serverAuth’ and another without. Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server and show that the X.509 certificate without the EKU fails.
Findings: PASS

461 Test 3: The evaluator shall send a server certificate in the TLS connection that does not match the server-selected ciphersuite (for example, send an ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite). The evaluator shall verify that the TOE disconnects after receiving the server’s Certificate handshake message.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server using any of the claimed ciphersuites. The Lightship TLS server will send back an otherwise validly constructed server certificate which does not match the requested the ciphersuite.
Findings: PASS

462 Test 4: The evaluator shall perform the following 'negative tests':

a) The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server using the TLS_NULL_WITH_NULL_NULL (cipher ID 0x0000).
Findings: PASS

b) Modify the server’s selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server sending a non-negotiated ciphersuite.
Findings: PASS

c) [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension the evaluator shall configure the server to perform an ECDHE or DHE key exchange in the TLS connection using a non-supported curve/group (for example P-192) and shall verify that the TOE disconnects after receiving the server’s Key Exchange handshake message.

High-Level Test Description
Force the TOE client to connect to a Lightship TLS server which will use an unsupported EC curve.

High-Level Test Description
Findings: PASS

463 Test 5: The evaluator performs the following modifications to the traffic:

a. Change the TLS version selected by the server in the Server Hello to a non-supported TLS version and verify that the client rejects the connection.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server advertising an incorrect TLS version.
Findings: PASS

b. [conditional]: If using DHE or ECDH, modify the signature block in the Server's Key Exchange handshake message, and verify that the handshake does not finished successfully, 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.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server sending a mangled key exchange signature.
Findings: PASS

464 Test 6: The evaluator performs the following 'scrambled message tests':

a. Modify a byte in the Server Finished handshake message and verify that the handshake does not finish successfully and no application data flows.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server sending a mangled finished message.
Findings: PASS

b. Send a garbled message from the server after the server has issued the ChangeCipherSpec message and verify that the handshake does not finish successfully and no application data flows.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server sending a mangled finished message.
Findings: PASS

c. Modify at least one byte in the server's nonce in the Server Hello handshake message and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.

High-Level Test Description
Using a Lightship developed TLS server, force the TOE client to attempt a handshake with a test server sending a modified nonce value. Do this once for a non-DHE ciphersuite and once for a DHE or ECDHE key exchange ciphersuite.
Findings: PASS

FCS_TLSC_EXT.1.2

465 Note that the following tests are marked conditional and are applicable under the following conditions:

a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.

or

b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable

or

c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.

466 Note that for some tests additional conditions apply.

467 IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:

- IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.
- IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.

468 The evaluator shall configure the reference identifier per the AGD guidance and perform the following tests during a TLS connection:

Note The TOE does not provide for, nor claim, any administrator-defined override mechanism for validating that the reference identifier matches that on the certificates for claimed TLS channels. Therefore, all of the following tests are applicable in the context of FCS_TLSC_EXT.1.

a. Test 1 [conditional]: 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. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the CN.

   Remark: 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.

High-Level Test Description
Force the TOE client to attempt a handshake with an OpenSSL s_server sub-application sending X.509 certificates that have the characteristics required by the test. Verify the TLS connections fails.
Findings: PASS

b. Test 2 [conditional]: 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 (e.g. IPv4, IPv6, FQDN, URI). When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the SAN.

High-Level Test Description
Force the TOE client to attempt a handshake with an OpenSSL s_server sub-application sending X.509 certificates that have the characteristics required by the test. Verify the TLS connection fails.
Findings: PASS

c. 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. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.

Findings:	The TOE mandates the presence of the SAN extension.

d. Test 4 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, SRV).

High-Level Test Description
Force the TOE client to attempt a handshake with an OpenSSL s_server sub-application sending X.509 certificates that have the characteristics required by the test. Verify the TLS connection succeeds.
Findings: PASS

e. Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID):

a) [conditional]: 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.

High-Level Test Description
Force the TOE client to attempt a handshake with an OpenSSL s_server sub-application sending X.509 certificates that have the characteristics required by the test. Verify the TLS connection fails.
Findings: PASS

b) [conditional]: 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 succeeds, if wildcards are supported, or fails if wildcards are not supported. 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.com) and verify that the connection fails. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)

High-Level Test Description
Force the TOE client to attempt a handshake with an OpenSSL s_server sub-application sending X.509 certificates that have the characteristics required by the test. Verify the TLS connection fails.
Findings: PASS

[Updated per TD 0634]

469 Objective: The objective of this test is to ensure the TOE is able to differentiate between IP address identifiers that are not allowed to contain wildcards and other types of identifiers that may contain wildcards.

f. Test 6: [conditional] If IP address identifiers supported in the SAN or CN, the evaluator shall present a server certificate that contains a CN that matches the reference identifier, except one of the groups has been replaced with a wildcard asterisk (*) (e.g. CN=*.168.0.1 when connecting to 192.168.0.1...

This negative test corresponds to the following section of the Application Note 64/105: "The exception being, the use of wildcards is not supported when using IP address as the reference identifier."

High-Level Test Description
Force the TOE client to attempt a handshake with an OpenSSL s_server sub-application sending X.509 certificates that have the characteristics required by the test. Verify the TLS connection fails.
Findings: PASS

g. Test 7 [conditional]: If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator modifies each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier):

5) The evaluator shall present a server certificate that does not contain an identifier in the Subject (DN) attribute type(s) that matches the reference identifier. The evaluator shall verify that the connection fails.

6) The evaluator shall present a server certificate that contains a valid identifier as an attribute type other than the expected attribute type (e.g. if the TOE is configured to expect id-at-serialNumber=correct_identifier, the certificate could instead include id-at-name=correct_identifier), and does not contain the SAN extension. The evaluator shall verify that the connection fails. Remark: 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 this test.

7) The evaluator shall present a server certificate that contains a Subject attribute type that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.

8) The evaluator shall confirm that all use of wildcards results in connection failure regardless of whether the wildcards are used in the left or right side of the presented identifier. (Remark: Use of wildcards is not addressed within RFC 5280.)

Findings:	The ST does not claim FPT_ITT.1 with RFC 5280.

FCS_TLSC_EXT.1.3

470 The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

471 Test 1: Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds, and a trusted channel can be established.

Findings:	This test case is performed as part of FIA_X509_EXT.1/Rev (EST).

472 Test 2: The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g. trusted channel was established) covering the types of failure for which an override mechanism is defined.

Findings:	This test case is performed as part of FIA_X509_EXT.1/Rev (EST). Appropriate override mechanisms are verified.

473 Test 3 [conditional]: The purpose of this test to verify that only selected certificate validation failures could be administratively overridden. If any override mechanism is defined for failed certificate validation, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative override available to accept such certificate.

Findings:	This test case is performed as part of FIA_X509_EXT.1/Rev (EST). Appropriate override mechanisms are verified.

FCS_TLSC_EXT.1.4

474 Test 1 [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension, the evaluator shall configure the server to perform ECDHE or DHE (as applicable) 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.

High-Level Test Description
Initiate a connection to the Lightship TLS server using each curve/group supported by the TOE for ECDHE key exchange. Verify the TLS connections succeed. Note that no DHE groups are claimed by the TOE.
Findings: PASS

4.2.8 FCS_TLSS_EXT.1 Extended: TLS Server Protocol

4.2.8.1 TSS

FCS_TLSS_EXT.1.1

475 The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component.

Findings:	[ST] / TOE Summary Specification lists the following ciphersuites: ■ TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

■ TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
■ TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
■ TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
■ TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
■ TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
■ TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
■ TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
■ TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
■ TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
■ TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
■ TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

The evaluator confirmed the ciphersuites were identical to those listed for this component.

FCS_TLSS_EXT.1.2

476 The evaluator shall verify that the TSS contains a description of how the TOE technically prevents the use of old SSL and TLS versions.

Findings:	[ST] / TOE Summary Specification states, “Only TLS 1.2 is supported. All connection attempts from remote clients requesting SSL2.0, SSL3.0, TLS1.0, or TLS 1.1 are denied.”

FCS_TLSS_EXT.1.3

[Updated per TD 0635]

477 If using ECDHE and/or DHE ciphers, the evaluator shall verify that the TSS lists all EC Diffie-Hellman curves and/or Diffie-Hellman groups used in the key establishment by the TOE when acting as a TLS Server. For example, if the TOE supports TLS_DHE_RSA_WITH_AES_128_CBC_SHA cipher and Diffie-Hellman parameters with size 2048 bits, then list Diffie-Hellman Group 14.

Findings:	[ST] / TOE Summary Specification states, “For TLS Server key establishment, if TLS_ECDHE_* ciphersuites are configured, the Security Administrator has the ability to also configure one of the following named curves and inclusive key exchange parameter: ■ secp256r1 NIST curve with 256-bit ECDHE ephemeral key agreement parameter for server key exchange which has at a minimum a 128-bit level of security. ■ secp384r1 NIST curve with 384-bit ECDHE ephemeral key agreement parameter for server key exchange which has at a minimum a 192-bit level of security.” “If a named curve is not configured, the secp256r1 NIST elliptic curve will be used by default.”

FCS_TLSS_EXT.1.4

478 The evaluator shall verify that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246) and/or if session resumption based on session tickets is supported (RFC 5077).

Findings:	[ST] / TOE Summary Specification states, “The TOE supports session resumption based on session tickets according to RFC 5077. The tickets adhere to the structural format provided in section 4 of RFC 5077.”

479 If session tickets are supported, the evaluator shall verify that the TSS describes that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption. The evaluator shall verify that the TSS identifies the key lengths and algorithms used to protect session tickets.

Findings:	[ST] / TOE Summary Specification – **"**The TOE supports session resumption based on session tickets according to RFC 5077. The tickets adhere to the structural format provided in section 4 of RFC 5077." "Session tickets are encrypted using 128-bit AES in CBC mode, which is consistent with FCS_COP.1/DataEncryption."

480 If session tickets are supported, the evaluator shall verify that the TSS describes that session tickets adhere to the structural format provided in section 4 of RFC 5077 and if not, a justification shall be given of the actual session ticket format.

Findings:	[ST] / TOE Summary Specification states, "The TOE supports session resumption based on session tickets according to RFC 5077. The tickets adhere to the structural format provided in section 4 of RFC 5077."

Note:	Updated per TD0569.

481 If the TOE claims a (D)TLS server capable of session resumption (as a single context, or across multiple contexts), the evaluator verifies that the TSS describes how session resumption operates (i.e. what would trigger a full handshake, e.g. checking session status, checking Session ID, etc.). If multiple contexts are used the TSS describes how session resumption is coordinated across those contexts. In case session establishment and session resumption are always using a separate context, the TSS shall describe how the contexts interact with respect to session resumption (in particular regarding the session ID). It is acceptable for sessions established in one context to be resumable in another context.

Findings:	[ST] / TOE Summary Specification state, "The TOE supports session resumption based on session tickets according to RFC 5077. The tickets adhere to the structural format provided in section 4 of RFC 5077. For FCS_TLSS_EXT.1, the TOE supports session resumption as a single context only. An encrypted session ticket containing the current session key information is sent by the TOE at the end of the TLS handshake. A web-client supporting session tickets will cache the ticket and may resume the earlier session by sending the encrypted session ticket in the handshake message. The TOE will decrypt the ticket, obtain the session key, and resume the session. Session tickets are encrypted using 128-bit AES in CBC mode, which is consistent with FCS_COP.1/DataEncryption."

4.2.8.2 Guidance Documentation

FCS_TLSS_EXT.1.1

482 The evaluator shall check the guidance documentation to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements).

Findings:	The HTTPS subsection of the Remote Administration Protocols subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] provides instructions for configuring TLS Server functionality on the TOE. The instructions are consistent with the description given in the TSS. The subsection states, "10. Configure HTTPS for the ciphersuite configuration option a. If you selected the Suite B ciphersuite enter: WLC(config)# ip http secure-ciphersuite ecdhe-ecdsa-aes-gcm-sha2 b. If not using the Suite B ciphersuite, you can choose any or all of the following configuration options for non-Suite B ciphersuites: WLC(config)# ip http secure-ciphersuite ecdhe-rsa-aes-cbc-sha2 ecdhe-rsa-aes-gcm-sha2 rsa-aes-cbc-sha2 rsa-aes-gcm-sha2 aes-128-cbc-sha aes-256-cbc-sha Refer to the HTTPS ciphersuite tables in this section for each configuration option and the supported ciphersuites."

FCS_TLSS_EXT.1.2

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

Findings:	The HTTPS subsection of the Remote Administration Protocols subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] provides instructions for configuring TLS Server functionality on the TOE. The subsection states, "9. Allow TLS v1.2 and deny TLS 1.1 and all lower versions WLC(config)# ip http tls-version TLSv1.2"

FCS_TLSS_EXT.1.3

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

Findings:	The HTTPS subsection of the Remote Administration Protocols subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] provides instructions on how to configure TLS key establishment so that it is consistent with the requirement. The subsection states, "2. If you chose the Suite B ciphersuite you must generate an elliptic curve key. Assign a label such as HTTPS-KEY WLC(config)# crypto key generate ec keysize [256 | 384] exportable label HTTPS-KEY Else if you want to use any of the non-Suite B ciphersuites you must generate a rsa key. Assign a label such as HTTPS-KEY WLC(config)# crypto key generate rsa general modulus [2048 | 3072] label HTTPS-KEY"

	And “11. If you would like to set the NIST elliptic curve use the following command. The choices are secp256r1 or secp384r1 and only one can be chosen. If you do not provide one the default NIST elliptic curve of secp256r1 will be used. Note: NIST elliptic curve does not apply to dhe-aes-cbc-sha2, dhe-aes-gcm-sha2, rsa-aes-cbc-sha2, rsa-aes-gcm-sha2. WLC(config)# ip http secure-ecdhe-curve <secp256r1 | secp384r1>”

NOTE:	Updated per TD0569.

FCS_TLSS_EXT.1.4

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

Findings:	The [AGD] does not identify any configuration requirements to support TLS session resumption based on session tickets according to RFC 5077, as is selected in the requirement.

4.2.8.3 Tests

FCS_TLSS_EXT.1.1

486 Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites 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 HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

High-Level Test Description
Using a Lightship developed TLS client, connect to the TOE using the claimed ciphersuites. Verify the TLS connections succeed.
Findings: PASS

487 Test 2: The evaluator shall send a Client Hello to the server with a list of ciphersuites that does not contain any of the ciphersuites in the server’s ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the server denies the connection.

High-Level Test Description
Using a Lightship developed TLS client, connect to the TOE using an unsupported ciphersuite. Then connect to the TOE using TLS_NULL_WITH_NULL_NULL. Verify each TLS connection fails.
Findings: PASS

488 Test 3: The evaluator shall perform the following modifications to the traffic:

a. Modify a byte in the Client Finished handshake message, and verify that the server rejects the connection and does not send any application data.

High-Level Test Description
Using a Lightship developed TLS client, connect to the TOE and modify the first payload byte in the Client Finished message. Verify the TLS connection fails.
Findings: PASS

b. (Test Intent: The intent of this test is to ensure that the server's TLS implementation immediately makes use of the key exchange and authentication algorithms to: a) Correctly encrypt (D)TLS Finished message and b) Encrypt every (D)TLS message after session keys are negotiated.)

The evaluator shall use one of the claimed ciphersuites to complete a successful handshake and observe transmission of properly encrypted application data. The evaluator shall verify that no Alert with alert level Fatal (2) messages were sent.

The evaluator shall verify that the Finished message (Content type hexadecimal 16 and handshake message type hexadecimal 14) is sent immediately after the server's ChangeCipherSpec (Content type hexadecimal 14) message. The evaluator shall examine the Finished message (encrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 11 22 33 44 55...) and confirm that it does not contain unencrypted data (unencrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 14 00 00 0c...), by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14 for at least one of three test messages. There is a chance that an encrypted Finished message contains a hexadecimal value of '14' at the position where a plaintext Finished message would contain the message type code '14'. If the observed Finished message contains a hexadecimal value of '14' at the position where the plaintext Finished message would contain the message type code, the test shall be repeated three times in total. In case the value of '14' can be observed in all three tests it can be assumed that the Finished message has indeed been sent in plaintext and the test has to be regarded as 'failed'. Otherwise it has to be assumed that the observation of the value '14' has been due to chance and that the Finished message has indeed been sent encrypted. In that latter case the test shall be regarded as 'passed'.

High-Level Test Description
Perform a successful handshake using one of the accepted ciphersuites and verify that the Server Finished message is encrypted.
Findings: PASS

FCS_TLSS_EXT.1.2

489 The evaluator shall send a Client Hello requesting a connection for all mandatory and selected protocol versions in the SFR (e.g. by enumeration of protocol versions in a test client) and verify that the server denies the connection for each attempt.

High-Level Test Description
Using a Lightship developed TLS client, connect to the TOE and attempt to negotiate SSL 2.0, SSL 3.0, TLS 1.0 and any unsupported, but otherwise valid TLS protocol versions contained in the PP. Verify such attempts fail.
Findings: PASS

FCS_TLSS_EXT.1.3

490 Test 1: [conditional] If ECDHE ciphersuites are supported:

a) The evaluator shall repeat this test for each supported elliptic curve. The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single supported elliptic curve specified in the Elliptic Curves Extension. The Evaluator shall verify (though a packet capture or instrumented client) that the TOE selects the same curve in the Server Key Exchange message and successfully establishes the connection.

High-Level Test Description
Using a Lightship developed TLS client, connect to the TOE using a valid ECDHE ciphersuite and curve combination and verify that the public key size that comes back in the Server Key Exchange message matches the expected bit size for the chosen curve.
Findings: PASS

b) The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single unsupported elliptic curve (e.g. secp192r1 (0x13)) specified in RFC4492, chap. 5.1.1. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.

High-Level Test Description
Using a Lightship developed TLS client, connect to the TOE using a valid ECDHE ciphersuite and an unsupported curve and verify that the TOE fails to send back a Server Hello message and terminates the connection.
Findings: PASS

491 Test 2: [conditional] If DHE ciphersuites are supported, the evaluator shall repeat the following test for each supported parameter size. If any configuration is necessary, the evaluator shall configure the TOE to use a supported Diffie-Hellman parameter size. The evaluator shall attempt a connection using a supported DHE ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Exchange Message where p Length is consistent with the message are the ones configured Diffie-Hellman parameter size(s).

Findings:	DHE ciphersuites are not claimed by the ST in the evaluated configuration.

492 Test 3: [conditional] If RSA key establishment ciphersuites are supported, the evaluator shall repeat this test for each RSA key establishment key size. If any configuration is necessary, the evaluator shall configure the TOE to perform RSA key establishment using a supported key size (e.g. by loading a certificate with the appropriate key size). The evaluator shall attempt a connection using a supported RSA key establishment ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a certificate whose modulus is consistent with the configured RSA key size.

High-Level Test Description
Using a Lightship developed TLS client, connect to the TOE using a valid pure RSA ciphersuite and verify that the certificate that comes back from the Server Certificate message matches the expected bit size.
Findings: PASS

FCS_TLSS_EXT.1.4

493 Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption).

494 Test 1 [conditional]: If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test:

a) The client sends a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket.
b) The client verifies the server does not send a NewSessionTicket handshake message (at any point in the handshake).
c) The client verifies the Server Hello message contains a zero-length session identifier or passes the following steps:
Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID.
d) The client completes the TLS handshake and captures the SessionID from the ServerHello.
e) The client sends a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d).
f) The client verifies the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

NOTE:	Updated per TD0569.

495 Remark: If multiple contexts are supported for session resumption, the session ID or session ticket may be obtained in one context for resumption in another context. It is possible that one or more contexts may only permit the construction of sessions to be reused in other contexts but not actually permit resumption themselves. For contexts which do not permit resumption, the evaluator is required to verify this behaviour subject to the description provided in the TSS. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

Findings:	The TOE supports session resumption based on session tickets according to RFC 2077.

496 Test 2 [conditional]: If the TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2), the evaluator shall carry out

the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

a) The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246).

b) The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

NOTE:	Updated per TD0569.

497 Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ID may be obtained in one context for resumption in another context. There is no requirement that the session ID be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ID constructed in another context must be tested. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

Findings:	The TOE does not claim support for session resumption using session IDs.

498 Test 3 [conditional]: If the TOE supports session tickets according to RFC5077, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

NOTE:	Test 3a modified per TD0555 and TD0556.

a) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE responds with an abbreviated handshake described in section 3.1 of RFC 5077 and illustrated with an example in figure 2. Of particular note: if the server successfully verifies the client's ticket, then it may renew the ticket by including a NewSessionTicket handshake message after the ServerHello in the abbreviated handshake (which is shown in figure 2). This is not required, however as further clarified in section 3.3 of RFC 5077.

b) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then

modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data.

NOTE:	Updated per TD0569.

499 Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ticket may be obtained in one context for resumption in another context. There is no requirement that the session ticket be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ticket constructed in another context must be tested. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.

High-Level Test Description
Show that the TOE will handle session resumption via Session Tickets as per the provided test steps. Using the Lightship TLS client for this test case, successfully resume a TLS session with the TOE using a valid session ticket. Verify the TOE performs an abbreviated handshake to resume the session. Using the Lightship TLS client for this test case, attempt resume a TLS session with the TOE using a modified session ticket. Verify the TOE rejects the session ticket, does not perform an abbreviated handshake, or resume the session.
Findings: PASS

4.3 Identification and Authentication (FIA)

4.3.1 FIA_X509_EXT.1/Rev X.509 Certificate Validation

4.3.1.1 TSS

500 The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied). It is expected that revocation checking is performed when a certificate is used in an authentication step and when performing trusted updates (if selected). It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected).

Findings:	[ST] / TOE Summary Specification states, “The TSF determines the validity of certificates by ensuring that the certificate and the certificate path are valid in accordance with RFC 5280. The certificate path is validated by ensuring that all the CA certificates have the basicConstraints extension and the CA flag is set to TRUE and the certificate path must terminate with a trusted CA certificate.” “The TOE ensures the extendedKeyUsage field includes: ■ The Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) for server certificates used in TLS. ■ The OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) for OCSP certificates used for OCSP responses.”

	"Revocation checking is performed on the leaf and intermediate certificate(s) when authenticating a certificate chain provided by the remote peer."

501 The TSS shall describe when revocation checking is performed and on what certificates. If the revocation checking during authentication is handled differently depending on whether a full certificate chain or only a leaf certificate is being presented, any differences must be summarized in the TSS section and explained in the Guidance.

Findings:	[ST] / TOE Summary Specification states, "Revocation checking is performed on the leaf and intermediate certificate(s) when authenticating a certificate chain provided by the remote peer. There are no functional differences if a full certificate chain or only a leaf certificate is presented."

4.3.1.2 Guidance Documentation

502 The evaluator shall also ensure that the guidance documentation describes where the check of validity of the certificates takes place, describes any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) and describes how certificate revocation checking is performed and on which certificate.

Findings:	[AGD] describes that certificate validity is checked in TLS and IPsec connections, and states: "Note: The TOE uses X.509v3 certificates to support authentication for IPsec connections. The TSF determines the validity of certificates by ensuring that the certificate and the certificate path are valid in accordance with RFC 5280. The certificate path is validated by ensuring that all the CA certificates have the basicConstraints extension and the CA flag is set to TRUE and the certificate path must terminate with a trusted CA certificate. OCSP is not supported; therefore the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) is trivially satisfied by the TOE."

4.3.1.3 Tests (RadSec)

503 The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols:

a. Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store)

   Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the

leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.

High-Level Test Description
Create a sequence of three valid X.509 certificates for RADIUS server communications as described in the AGD: a self-signed root CA, an intermediate CA signed by the root CA and a leaf node certificate signed by the intermediate CA. Load the chain into the TOEs trust store. Force the TOE to connect to the OpenSSL TLS server, posing as the RADIUS server, that sends back a Server Certificate message and show that the connection is accepted. Remove the root CA from the TOE trust store. Force the TOE to connect to the OpenSSL TLS server show that the connection is no longer accepted.
Findings: PASS

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

High-Level Test Description
Force the TOE to connect to the OpenSSL TLS server that sends back an expired server certificate and show it is not accepted. Force the TOE to connect to the OpenSSL TLS server that sends back a valid server certificate and show the connection succeeds. Wait for the intermediate certificate to expire and force the TOE to connect to the OpenSSL TLS again. Show the connection fails. Show the certificate chain validation fails.
Findings: PASS

c. Test 3: The evaluator shall test that the TOE can properly handle revoked certificates-–conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the peer certificate and revocation of the peer intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.

High-Level Test Description
Configure CRL responder for RADIUS certificate validation with empty CRLs for the intermediate and root certificates. Initiate a connection from the TOE to the OpenSSL TLS server. Verify the CRLs are queried by the TOE and the connection succeeds. Revoke the RADIUS server certificate and initiate a connection from the TOE to the OpenSSL TLS server. verify the TOE queries the CRL server and that the connection now fails due to the certificate being revoked. Un-revoke the RADIUS server certificate, revoke the intermediate certificate and initiate a connection from the TOE to the OpenSSL TLS server. Verify the TOE queries the CRL server and that the connection now fails due to the certificate being revoked.
Findings: PASS

d. Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

High-Level Test Description
The ST does not claim OCSP support. Ensure the intermediate certificate on the TOE is replaced with one lacking the cRLsign key usage bit and initiate a connection from the TOE to the OpenSSL TLS server. Verify the CRL validation fails.
Findings: PASS

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

High-Level Test Description
Force the TOE to connect to a Lightship test server which will send back a properly mangled X.509 certificate in which the ASN.1 header bytes in the first 8 bytes are modified. Verify the certificate fails to validate.
Findings: PASS

f. Test 6: The evaluator shall modify any byte in the certificate signatureValue field (see RFC5280 Sec. 4.1.1.3), which is normally the last field in the certificate, and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

High-Level Test Description
Force the TOE to connect to a Lightship test server which will send back an X.509 certificate in which the last byte of the server certificate (the signature) is modified.
Findings: PASS

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

High-Level Test Description
Force the TOE to connect to a Lightship test server which will send back an X.509 certificate in which the public key of the server certificate is modified. Verify the certificate fails to validate.
Findings: PASS

NOTE:	Update per TD0527.

504 Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The evaluator shall conduct the following tests:

Test 8a: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates (terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

Findings:	The TOE does not support EC certificates for RADIUS communications.

Test 8b: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

Findings:	The TOE does not support EC certificates for RADIUS communications.

Test 8c: The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store.

Findings:	The TOE does not support EC certificates for RADIUS communications.

505 The evaluator shall perform the following tests for FIA_X509_EXT.1.2/Rev. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/Rev. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted.

506 The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation).

507 For each of the following tests the evaluator shall create a chain of at least three certificates: a self-signed root CA certificate, an intermediate CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain).

a. Test 1: The evaluator shall ensure that at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

High-Level Test Description
Attempt to replace the known-good intermediate CA on the TOE with a cloned copy that has no Basic Constraints extension. Verify the certificate fails to upload to the TOE's trust store.
Findings: PASS

b. Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

High-Level Test Description
Attempt to replace the known-good intermediate CA on the TOE with a cloned copy that has the CA flag set to false in the Basic Constraints extension. Verify the certificate fails to upload to the TOE's trust store.
Findings: PASS

508 The evaluator shall repeat these tests for each distinct use of certificates. Thus, for example, use of certificates for TLS connection is distinct from use of certificates for trusted updates so both of these uses would be tested. But there is no need to repeat the tests for each separate TLS channel in FTP_ITC.1 and FTP_TRP.1/Admin (unless the channels use separate implementations of TLS).

Findings:	Tests were repeated for distinct use of certificates in RadSec, EST and IPsec connections. Test verdicts found below.

4.3.1.4 Tests (EST)

509 The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols:

a. Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store)

   Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.

High-Level Test Description
Create a sequence of three valid X.509 certificates for EST server communications as described in the AGD: a self-signed root CA, an intermediate CA signed by the root CA and a leaf node certificate signed by the intermediate CA. Load the chain into the TOEs trust store. Force the TOE to connect to the OpenSSL TLS server, posing as the EST server. Verify the server sends the expected certificate in the Server Certificate message and show that the connection is accepted. Remove the root CA from the TOE trust store. Force the TOE to connect to the OpenSSL TLS server show that the connection is no longer accepted.
Findings: PASS

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

High-Level Test Description
Force the TOE to connect to the OpenSSL TLS server that sends back an expired server certificate and show it is not accepted. Force the TOE to connect to the OpenSSL TLS server that sends back a valid server certificate and show the connection succeeds. Wait for the intermediate certificate to expire and force the TOE to connect to the OpenSSL TLS again. Show the connection fails. Show the certificate chain validation fails.
Findings: PASS

c. Test 3: The evaluator shall test that the TOE can properly handle revoked certificates-–conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the peer certificate and revocation of the peer intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.

High-Level Test Description
The TOE does not claim OCSP support. Configure a CRL responder for EST certificate validation with empty CRLs for the intermediate and root Cas. Initiate a connection from the TOE to the OpenSSL TLS server. Verify the CRLs are queried by the TOE and the connection succeeds. Revoke the EST server certificate and initiate a connection from the TOE to the OpenSSL TLS server. Verify the TOE queries the CRL server and that the connection now fails due to the certificate being revoked. Un-revoke the EST server certificate, revoke the intermediate certificate and initiate a connection from the TOE to the OpenSSL TLS server. Verify the TOE queries the CRL server and that the connection now fails due to the certificate being revoked.
Findings: PASS

d. Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

High-Level Test Description
The ST does not select OCSP functionality. Ensure the intermediate certificate on the TOE is replaced with a valid copy lacking the cRLsign key usage bit and initiate a connection from the TOE to the OpenSSL TLS server. Verify the CRL validation fails.
Findings: PASS

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

High-Level Test Description
Force the TOE to connect to a Lightship test server which will send back a properly mangled X.509 certificate in which the ASN.1 header bytes in the first 8 bytes are modified. Verify the certificate fails to validate.

High-Level Test Description
Findings: PASS

f. Test 6: The evaluator shall modify any byte in the certificate signatureValue field (see RFC5280 Sec. 4.1.1.3), which is normally the last field in the certificate, and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

High-Level Test Description
Force the TOE to connect to a Lightship test server which will send back an X.509 certificate in which the last byte of the certificate (the signature) is modified. Verify the certificate fails to validate.
Findings: PASS

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

High-Level Test Description
Force the TOE to connect to a Lightship test server which will send back an X.509 certificate in which the public key of the certificate is modified. Verify the certificate fails to validate.
Findings: PASS

NOTE:	Update per TD0527.

510 Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The evaluator shall conduct the following tests:

Test 8a: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates (terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

Findings:	The TOE does not claim the ability to process CA certificates presented in certificate messages.

Test 8b: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC

certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

Findings:	The TOE does not claim the ability to process CA certificates presented in certificate messages.

Test 8c: The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store.

High-Level Test Description
Construct a chain of three ECDSA certificates: a leaf, an intermediate CA and a trust anchor. Create a clone of the Intermediate CA, such that the public key is explicitly defined rather than being a named curve. Verify the named curve version of the intermediate certificate can be successfully uploaded/validated. Remove the named curve version of the intermediate certificate from the TOE's trust store. Verify the explicit format intermediate cannot be successfully uploaded/validated.
Findings: PASS

511 The evaluator shall perform the following tests for FIA_X509_EXT.1.2/Rev. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/Rev. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted.

512 The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation).

513 For each of the following tests the evaluator shall create a chain of at least three certificates: a self-signed root CA certificate, an intermediate CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain).

a. Test 1: The evaluator shall ensure that at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store

(i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

Findings:	This test was covered as part of FIA_X509_EXT.1.2/Rev (RadSec tests) above.

b. Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

Findings:	This test was covered as part of FIA_X509_EXT.1.2/Rev (RadSec tests) above.

514 The evaluator shall repeat these tests for each distinct use of certificates. Thus, for example, use of certificates for TLS connection is distinct from use of certificates for trusted updates so both of these uses would be tested. But there is no need to repeat the tests for each separate TLS channel in FTP_ITC.1 and FTP_TRP.1/Admin (unless the channels use separate implementations of TLS).

Findings:	Tests were repeated for distinct use of certificates in RadSec, EST and IPsec connections.

4.3.1.5 Tests (IPsec)

515 The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols:

a. Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store)

   Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the

leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.

High-Level Test Description
Create a sequence of three X.509 certificates for Ipsec communications as described in the AGD: a root CA, an intermediate CA signed by the root CA and a leaf node certificate signed by the intermediate CA. Upload the certificate chain to the TOE's trust store. Initiate an Ipsec connection from the TOE to the test workstation. Show that the X.509 peer certificate validation using the recently uploaded certificate chain succeeds. Remove the root CA from the TOE trust store. Initiate an Ipsec connection from the TOE to the test workstation. Show that the X.509 peer certificate validation using the recently uploaded certificate chain fails.
Findings: PASS

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

High-Level Test Description
Initiate an IPsec connection from the TOE to the test workstation. Have the IPsec peer send back an expired server certificate. Show the connection fails. Show the certificate chain validation fails. Initiate an IPsec connection from the TOE to the test workstation. Have the IPsec peer send back a valid server certificate and show the connection succeeds. Wait for the intermediate certificate to expire and initiate an IPsec connection from the TOE to the test workstation. Show the connection fails. Show the certificate chain validation fails.
Findings: PASS

c. Test 3: The evaluator shall test that the TOE can properly handle revoked certificates-–conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the peer certificate and revocation of the peer intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.

High-Level Test Description
The ST does not claim OCSP support. Configure a CRL responder for IPsec certificate validation with empty CRLs for the intermediate and root CAs. Initiate a connection from the TOE to the IPsec peer. Verify the CRLs are queried by the TOE and the connection succeeds. Revoke the IPsec server certificate and initiate a connection from the TOE to the IPsec peer. Verify the TOE queries the CRL server and that the connection now fails due to the certificate being revoked.

High-Level Test Description
Un-revoke the IPsec server certificate, revoke the intermediate certificate and initiate a connection from the TOE to the IPsec peer. Verify the TOE queries the CRL server and that the connection now fails due to the certificate being revoked.
Findings: PASS

d. Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

High-Level Test Description
The ST does not claim OCSP functionality. Ensure the intermediate certificate used in Test 1 is replaced with a valid copy lacking the cRLsign key usage bit and initiate a connection from the TOE to the IPsec peer. Verify the CRL validation fails.
Findings: PASS

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

High-Level Test Description
Initiate an IPsec connection from the TOE to a modified IPsec peer using custom Lightship library functions for this test case which will send back a properly mangled X.509 server certificate in which the ASN.1 header bytes in the first 8 bytes are modified. Verify the server certificate fails to validate.
Findings: PASS

f. Test 6: The evaluator shall modify any byte in the certificate signatureValue field (see RFC5280 Sec. 4.1.1.3), which is normally the last field in the certificate, and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

High-Level Test Description
Initiate an IPsec connection from the TOE to a modified IPsec peer using custom Lightship library functions for this test case which will send back an X.509 certificate in which the last byte of the certificate (the signature) is modified.
Findings: PASS

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

High-Level Test Description
Initiate an IPsec connection from the TOE to a modified IPsec peer using custom Lightship library functions for this test case which will send back an X.509 certificate in which the public key of the certificate is modified.
Findings: PASS

NOTE:	Update per TD0527.

516 Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The evaluator shall conduct the following tests:

Test 8a: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates (terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

Findings:	The TOE does not claim the ability to process CA certificates presented in certificate messages.

Test 8b: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

Findings:	The TOE does not claim the ability to process CA certificates presented in certificate messages.

Test 8c: The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store.

Findings:	This test was covered in FIA_X509_EXT.1.1/Rev for EST Test 8c.

517 The evaluator shall perform the following tests for FIA_X509_EXT.1.2/Rev. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/Rev. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted.

518 The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation).

519 For each of the following tests the evaluator shall create a chain of at least three certificates: a self-signed root CA certificate, an intermediate CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain).

c. Test 1: The evaluator shall ensure that at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

Findings:	This test was covered in FIA_X509_EXT.1.2/Rev (RadSec).

d. Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

Findings:	This test was covered in FIA_X509_EXT.1.2/Rev (RadSec).

520 The evaluator shall repeat these tests for each distinct use of certificates. Thus, for example, use of certificates for TLS connection is distinct from use of certificates for trusted updates so both of these uses would be tested. But there is no need to repeat the tests for each separate TLS channel in FTP_ITC.1 and FTP_TRP.1/Admin (unless the channels use separate implementations of TLS).

Findings:	Tests were repeated for distinct use of certificates in RadSec, EST and IPsec connections.

4.3.2 FIA_X509_EXT.2 X.509 Certificate Authentication

4.3.2.1 TSS

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

Findings:	[ST] / TOE Summary Specification states, “The TOE determines which certificate to use based upon the trustpoint configured. The instructions for configuring trustpoints is provided in CC Configuration Guide.“

522 The evaluator shall examine the TSS to confirm that it describes the behaviour of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the guidance documentation contains instructions on how this configuration action is performed.

Findings:	[ST] / TOE Summary Specifications states, “In the event that a network connection cannot be established to verify the revocation status of certificate for an external peer the connection will be rejected. For internal TOE communication in accordance with FPT_ITT.1, certificate revocation checking is not performed.”

4.3.2.2 Guidance Documentation

523 The evaluator shall also ensure that the guidance documentation describes the configuration required in the operating environment so the TOE can use the certificates. The guidance documentation shall also include any required configuration on the TOE to use the certificates. The guidance document shall also describe the steps for the Security Administrator to follow if the connection cannot be established during the validity check of a certificate used in establishing a trusted channel.

Findings:	Certificates are used by the TOE for IPsec, DTLS, HTTPS and TLS (RADsec/EST). Instructions on how to configure the TOE to use certificates for each of these purposes are found in the associated section of the [AGD]. Namely, these instructions are found in the TLS-RADsec, DTLS-CAPWAP, CC Mode, IPsec and HTTPS subsections of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD]. The CRLs or OCSP Server Choosing a Certificate Revocation Mechanism chapter of reference document [12] in the [AGD] states, “If your device does not have the applicable CRL and is unable to obtain one or if the OCSP server returns an error, your device will reject the peer’s certificate--unless you include the none keyword in your configuration. If the none keyword is configured, a revocation check will not be performed and the certificate will always be accepted.”

4.3.2.3 Tests (RadSec)

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

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

High-Level Test Description
Initiate a connection from the TOE to the OpenSSL TLS server. Verify the TOE attempts to query the CRL server and fails. Verify the certificate verification relying on the CRLs fails.
Findings: PASS

4.3.2.4 Tests (EST)

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

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

High-Level Test Description
Initiate a connection from the TOE to the OpenSSL TLS server. Verify the TOE attempts to query the CRL server and fails. Verify the certificate verification relying on the CRLs fails.
Findings: PASS

4.3.2.5 Tests (IPsec)

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

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

High-Level Test Description
Initiate an IPsec connection from the TOE to the IPsec peer. Verify the TOE attempts to query the CRL server and fails. Verify the certificate verification relying on the CRLs fails.
Findings: PASS

4.3.3 FIA_X509_EXT.3 Extended: X509 Certificate Requests

4.3.3.1 TSS

530 If the ST author selects "device-specific information", the evaluator shall verify that the TSS contains a description of the device-specific fields used in certificate requests.

Findings:	The option for “device-specific information” was not selected.

4.3.3.2 Guidance Documentation

The evaluator shall check to ensure that the guidance documentation contains instructions on requesting certificates from a CA, including generation of a Certificate Request. If the ST author selects "Common Name", "Organization", "Organizational Unit", or "Country", the evaluator shall ensure that this guidance includes instructions for establishing these fields before creating the Certification Request.

Findings:	Instructions on how to generate certificate requests and requesting certificates from a CA are found in each section of the [AGD] that leverage X.509 certificates. Namely, these instructions are found in the TLS-RADsec, CC Mode, IPsec and HTTPS subsections of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD]. The evaluator confirmed these sections include instructions on establishing values for the Common Name, Organization, Organizational Unit, and Country fields prior to creation of a certificate request. Namely, these section state, “WLC(ca-trustpoint)# subject-name C=<two letter country code>, ST=<two letter state code>, L=<locality>, O=<organization>, OU=<organizational unit>, CN=<Common Name> For example: subject-name C=US, ST=MA, L=Boxborough, O=STO, OU=GCT, CN=C9800”

4.3.3.3 Tests

531 The evaluator shall perform the following tests:

a. Test 1: The evaluator shall use the guidance documentation to cause the TOE to generate a Certification Request. The evaluator shall capture the generated message and ensure that it conforms to the format specified. The evaluator shall confirm that the Certification Request provides the public key and other required information, including any necessary user-input information.

High-Level Test Description
Using the TOE CSR generator, create a new CSR and download to an external CA entity for signing. Using OpenSSL, verify that the information in the CSR is as expected.
Findings: PASS

b. Test 2: The evaluator shall demonstrate that validating a response message to a Certification Request without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the certificate response message, and demonstrate that the function succeeds.

High-Level Test Description
The CSR from the previous test is signed and reimported into the TOE. The certificate is then assigned a purpose, at which point the certificate is validated. If it cannot be validated, it cannot be assigned a purpose and therefore cannot be used. Create a cert chain consisting of a CA and Intermediate certificates. Use CA and Intermediate certificates to create valid trustpoints on the TOE. Do not “authenticate” Intermediate certificate. Use Intermediate trustpoint to generate a CSR. Sign the CSR with the unauthenticated intermediate certificate. Attempt to import the certificate into the TOE. Authenticate the intermediate certificate. Attempt to import the certificate into the TOE.
Findings: PASS

4.4 Security management (FMT)

4.4.1 FMT_MOF.1/Functions Management of security functions behaviour

4.4.1.1 TSS

532 For distributed TOEs see chapter 2.4.1.1. For distributed TOEs it is required to verify the TSS to ensure that it describes how every function related to security management is realized for every TOE component and shared between different TOE components. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs.

Findings:	[ST] / TOE Summary Specification states, “The WLC provides all the capabilities necessary to centrally manage all TOE components. There is no remote trusted path administrative interface available directly on the Access Points. In addition, the TOE prohibits direct Access Point administration on the local console.” “Only the authorized Administrator on the WLC may: ■ … ■ Modify the security function behavior including: o Transmission of audit data to an external syslog server; o Enable or disable logging to the local audit log, or to the local console, or to remote syslog servers, and to display the configuration and status of audit functions.”

533 For non-distributed TOEs, the evaluator shall ensure the TSS for each administrative function identified the TSS details how the Security Administrator determines or modifies the behaviour of (whichever is supported by the TOE) transmitting audit data to an external IT entity, handling of audit data, audit functionality when Local Audit Storage Space is full (whichever is supported by the TOE).

Findings:	The TOE is a distributed TOE.

4.4.1.2 Guidance Documentation

534 For distributed TOEs see chapter 2.4.1.2. For distributed TOEs it is required to verify the Guidance Documentation to describe management of each TOE component. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs.

Findings:	The evaluator confirmed the [AGD] describes how each function related to security management is realized for each TOE component and that all relevant aspects of each TOE component are covered by the FMT SFRs. The FIPS chapter of Reference Document [6] of the [AGD] states, “The console of APs get disabled when the controller is operating in FIPS mode.” [AGD] section “Access Remote Administrative Interfaces” describes the management of each TOE component. “Note: The WLC provides all the capabilities necessary to centrally manage all TOE components. There is no remote trusted path administrative interface available directly on the Access Points. In addition, the TOE prohibits direct Access Point administration on the local console.” The evaluator confirmed that all relevant aspects of each TOE component are covered by the FMT SFRs.

535 For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the Security Administrator determines or modifies the behaviour of (whichever is supported by the TOE) transmitting audit data to an external IT entity, handling of audit data, audit functionality when Local Audit Storage Space is full (whichever is supported by the TOE) are performed to include required configuration settings.

Findings:	The TOE is a distributed TOE.

4.4.1.3 Tests

536 Test 1 (if ‘transmission of audit data to external IT entity’ is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify all security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). Attempts to modify parameters without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to modify the security related parameters can be executed. In that case it shall be

demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

High-Level Test Description
Log into the TOE as a low privileged user. Attempt to change Syslog server settings. The attempt should fail.
Findings: PASS

537 Test 2 (if ‘transmission of audit data to external IT entity’ is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify all security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity with prior authentication as Security Administrator. The effects of the modifications should be confirmed.

538 The evaluator does not have to test all possible values of the security related parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity but at least one allowed value per parameter.

High-Level Test Description
Using the privileged ‘testadmin’ user modify the IP and port of the syslog provider. Verify that the TOE attempts to communicate using the new parameters.
Findings: PASS

539 Test 1 (if 'handling of audit data' is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify all security related parameters for configuration of the handling of audit data without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). Attempts to modify parameters without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator. The term ‘handling of audit data’ refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.2, FAU_STG_EXT.1.3 and FAU_STG_EXT.2/LocSpace.

Findings: The ST does not claim ‘handling of audit data’ together with ‘modify the behavior of’ for FMT_MOF.1/Functions.

540 Test 2 (if 'handling of audit data' is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify all security related parameters for configuration of the handling of audit data with prior authentication as Security Administrator. The effects of the modifications should be confirmed. The term ‘handling of audit data’ refers to the different options for selection and assignments in SFRs FAU_STG_EXT.1.2, FAU_STG_EXT.1.3 and FAU_STG_EXT.2/LocSpace.

541 The evaluator does not necessarily have to test all possible values of the security related parameters for configuration of the handling of audit data but at least one allowed value per parameter.

Findings:	The ST does not claim ‘handling of audit data’ together with ‘modify the behavior of’ for FMT_MOF.1/Functions.

542 Test 1 (if 'audit functionality when Local Audit Storage Space is full' is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify the behaviour when Local Audit Storage Space is full without prior authentication as Security Administrator (by authentication as_a user with no administrator privileges or without user authentication at all). This attempt should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

Findings:	The ST does not claim ‘audit functionality when Local Audit Storage Space is full’ together with ‘modify the behavior of’ for FMT_MOF.1/Functions.

543 Test 2 (if 'audit functionality when Local Audit Storage Space is full' is selected from the second selection together with 'modify the behaviour of' in the first selection): The evaluator shall try to modify the behaviour when Local Audit Storage Space is full with prior authentication as Security Administrator. This attempt should be successful. The effect of the change shall be verified.

544 The evaluator does not necessarily have to test all possible values for the behaviour when Local Audit Storage Space is full but at least one change between allowed values for the behaviour.

Findings:	The ST does not claim ‘audit functionality when Local Audit Storage Space is full’ together with ‘modify the behavior of’ for FMT_MOF.1/Functions.

545 Test 3 (if in the first selection 'determine the behaviour of' has been chosen together with for any of the options in the second selection): The evaluator shall try to determine the behaviour of all options chosen from the second selection without prior authentication as Security Administrator (by authentication as a user with no administrator privileges or without user authentication at all). This can be done in one test or in separate tests. The attempt(s) to determine the behaviour of the selected functions without administrator authentication shall fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

Findings:	The ST does not claim ‘determine the behaviour of’ in FMT_MOF.1/Functions.

546 Test 4 (if in the first selection 'determine the behaviour of' has been chosen together with for any of the options in the second selection): The evaluator shall try to determine the behaviour of all options chosen from the second selection with prior authentication as Security Administrator. This can be done in one test or in separate

tests. The attempt(s) to determine the behaviour of the selected functions with administrator authentication shall be successful.

Findings:	The ST does not claim ‘determine the behaviour of’ in FMT_MOF.1/Functions.

4.4.2 FMT_MOF.1/Services Management of Security Functions Behaviour

4.4.2.1 TSS

547 For distributed TOEs see chapter 2.4.1.1. For distributed TOEs it is required to verify the TSS to ensure that it describes how every function related to security management is realized for every TOE component and shared between different TOE components. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs.

Findings:	[ST] / TOE Summary Specification states, “The WLC provides all the capabilities necessary to centrally manage all TOE components. There is no remote trusted path administrative interface available directly on the Access Points. In addition, the TOE prohibits direct Access Point administration on the local console.” “Only the authorized Administrator on the WLC may: ■ … ■ Start and Stop Services;”

548 For non-distributed TOEs, the evaluator shall ensure the TSS lists the services the Security Administrator is able to start and stop and how that how that operation is performed.

Findings:	The TOE is a distributed TOE.

4.4.2.2 Guidance Documentation

549 For distributed TOEs see chapter 2.4.1.2.

Findings:	The evaluator confirmed the [AGD] describes how each function related to security management is realized for each TOE component and that all relevant aspects of each TOE component are covered by the FMT SFRs.

550 For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the TSS lists the services the Security Administrator is able to start and stop and how that how that operation is performed.

Findings:	The TOE is a distributed TOE.

4.4.2.3 Tests

551 The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) without prior authentication as Security Administrator (either by authenticating as a user with no administrator privileges, if possible, or without prior authentication at all). The attempt to enable/disable this service/these services should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to enable/disable this service/these services can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

High-Level Test Description
As a non-privileged admin, attempt to start/stop logging services through the CLI. As a non-privileged admin, attempt to start/stop logging services through the Web GUI.
Findings: PASS

552 The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) with prior authentication as Security Administrator. The attempt to enable/disable this service/these services should be successful.

High-Level Test Description
As a privileged admin, attempt to start/stop logging services through the CLI. As a privileged admin, attempt to start/stop logging services through the Web GUI.
Findings: PASS

4.4.3 FMT_MTD.1/CryptoKeys Management of TSF Data

4.4.3.1 TSS

553 For distributed TOEs see chapter 2.4.1.1. For distributed TOEs it is required to verify the TSS to ensure that it describes how every function related to security management is realized for every TOE component and shared between different TOE components. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs.

Findings:	[ST] / TOE Summary Specification states “The WLC provides all the capabilities necessary to centrally manage all TOE components. There is no remote trusted path administrative interface available directly on the Access Points. In addition, the TOE prohibits direct Access Point administration on the local console.” [ST] / TOE Summary Specification (FMT_MTD.1/CoreData) states “all Admin functions including those functions that manage TSF data are mediated by the TOE which ensures there is no capability to manage TSF data at any administrative interface until an administrator is successfully identified and authenticated. “In addition, the TOE ensures management of truststores (trustpoints) containing X.509 certificates is restricted to the authorized Administrator. User accounts with less than level 15 privilege do not have the ability to add or remove a truststore.”

554 For non-distributed TOEs, the evaluator shall ensure the TSS lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and how that how those operations are performed.

Findings:	The TOE is a distributed TOE.

4.4.3.2 Guidance Documentation

555 For distributed TOEs see chapter 2.4.1.2.

Findings:	The evaluator confirmed the [AGD] describes how each function related to security management is realized for each TOE component and that all relevant aspects of each TOE component are covered by the FMT SFRs.

556 For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and how that how those operations are performed.

Findings:	The TOE is a distributed TOE.

4.4.3.3 Tests

557 The evaluator shall try to perform at least one of the related actions (modify, delete, generate/import) without prior authentication as Security Administrator (either by authentication as a non-administrative user, if supported, or without authentication at all). Attempts to perform related actions without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to manage cryptographic keys can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.

558 The evaluator shall try to perform at least one of the related actions with prior authentication as Security Administrator. This attempt should be successful.

High-Level Test Description
As a non-privileged admin, attempt to generate a new key through the CLI. Verify the attempt fails. As a privileged admin, attempt to generate a new key through the CLI. As a non-privileged admin, attempt to generate a new key through the We GUI. Verify the attempt fails. As a privileged admin, attempt to generate a new key through the Web GUI. Verify the attempt succeeds. Verify the attempt succeeds.
Findings: PASS

5 Evaluation activities for WLAN Extended Profile

5.1 Cryptographic Support (FCS)

5.1.1 FCS_CKM.1/WPA2 - FCS_CKM.1(2) Cryptographic Key Generation (Symmetric Keys for WPA2 Connections)

5.1.1.1 TSS

559 The cryptographic primitives will be verified through assurance activities specified elsewhere in this EP. The evaluator shall verify that the TSS describes how the primitives defined and implemented by this EP are used by the TOE in establishing and maintaining secure connectivity to the wireless clients. This description shall include how the GTK and PTK are generated or derived. The TSS shall also provide a description of the developer's method(s) of assuring that their implementation conforms to the cryptographic standards; this includes not only testing done by the developing organization, but also proof of third-party testing that is performed (e.g. WPA2 certification). The evaluator shall ensure that the description of the testing methodology is of sufficient detail to determine the extent to which the details of the protocol specifics are tested.

Findings:	[ST] / TOE Summary Specification states, "The Authenticator and Supplicant perform a four-way handshake to derive the PTK and if necessary, the GTK temporal keys from the master keys. The TSF implements PRF-384 and PRF-704 key derivation algorithms as specified in [IEEE 802.11-2012] and [IEEE 802.11ac-2013] respectively, to derive the number of bits required to obtain Pairwise Transient Key (PTK) and Group Temporal Key (GTK) keys." "Certification testing performed by the Wi-Fi Alliance demonstrates the TOE implements the IEEE 802.11-2012 standard correctly. Refer to Table 24 for identification of the relevant Wi-Fi Alliance certificates."

5.1.1.2 Guidance Documentation

560 There are no Guidance assurance activities.

5.1.1.3 Tests

561 The evaluator shall also perform the following test using a packet sniffing tool to collect frames between the TOE and a wireless client:

562 Step 1: The evaluator shall configure the access point to an unused channel and configure the WLAN sniffer to sniff only on that channel (i.e., lock the sniffer on the selected channel). The sniffer should also be configured to filter on the MAC address of the TOE and/or client.

563 Step 2: The evaluator shall configure the TOE to communicate with a WLAN client using IEEE 802.11-2012 and a 256-bit (64 hex values 0-f) pre-shared key, setting up the connections as described in the operational guidance. The pre-shared key is only used for testing.

564 Step 3: The evaluator shall start the sniffing tool, initiate a connection between the TOE and WLAN client, and allow the TOE to authenticate, associate and successfully complete the 4-way handshake with the client.

565 Step 4: The evaluator shall set a timer for 1 minute, at the end of which the evaluator shall disconnect the client from the TOE and stop the sniffer.

566 Step 5: The evaluator shall identify the 4-way handshake frames (denoted EAPOL-key in Wireshark captures) and derive the PTK from the 4-way handshake frames and pre-shared key as specified in IEEE 802.11-2012.

567 Step 6: The evaluator shall select the first data frame from the captured packets that was sent between the client and TOE after the 4-way handshake successfully completed, and without the frame control value 0x4208 (the first 2 bytes are 08 42). The evaluator shall use the PTK to decrypt the data portion of the packet as specified in IEEE 802.11-2012, and shall verify that the decrypted data contains ASCII-readable text.

568 Step 7: The evaluator shall repeat Step 6 for the next 2 data frames between the TOE and client, and without frame control value 0x4208.

High-Level Test Description
Sniff the wireless traffic between the AP and a wireless client and initiate a client connection to the test WLAN. Examine the captured wireless frames and verify all 4 EAPOL handshake messages are captured, and that the client authentication is successful. Let the capture run for 1 minute after authentication takes place and send traffic through the wireless channel to generate data frames. Use wireshark's built in IEEE802.11 WPA decryption functionality to decrypt 3 data frames using the PTK. Verify the data contains ASCII readable text.
Findings: PASS

Technical Decision: This test was modified per TD0282.

569 Additionally, the evaluator shall test the PRF function using the test vectors from:

- Section 2.4 "The PRF Function – PRF(key, prefix, data, length)" of the IEEE 802.11-02/362r6 document "Proposed Test vectors for IEEE 802.11 TGi" dated September 10, 2002, and
- Annex M.3 "PRF reference implementation and test vectors" of IEEE 802.11-2012.

Findings: See Table 24 in the [ST] for Wifi Alliance certificates.

5.1.2 FCS_CKM.2/PMK – FCS_CKM.2(2) Cryptographic Key Distribution (PMK)

5.1.2.1 TSS

570 The evaluator shall examine the TSS to determine that it describes how the PMK is transferred (that is, through what EAP attribute) to the TSF.

Findings: [ST] / TOE Summary Specification states, "The TOE provides RADsec to protect the PMK received from the RADIUS authentication server. The PMK is received by the TOE (Authenticator) via the MS-MPPE-Recv-Key EAP attribute."

5.1.2.2 Guidance Documentation

571 There are no Guidance assurance activities.

5.1.2.3 Tests

572 The evaluator shall establish a session between the TOE and a RADIUS server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the RADIUS server and the TOE during a successful attempt to connect a wireless client to the TOE to determine that the PMK is not exposed.

High-Level Test Description
With the WLC in the evaluated configuration, attempt to associate/authenticate a wireless client using EAP-TLS. Verify the wireless connection is successful. Inspect the traffic between the WLC and the RADIUS server to determine that the PMK is not exposed.
Findings: PASS

5.1.3 FCS_CKM.2/GTK – FCS_CKM.2(3) Cryptographic Key Distribution (GTK)

5.1.3.1 TSS

573 The evaluator shall check the TSS to ensure that it describes how the GTK is wrapped prior to be distributed using the AES implementation specified in this EP, and also how the GTKs are distributed when multiple clients connect to the TOE.

Findings:	[ST] / TOE Summary Specification states, "The Authenticator securely distributes the GTK to the Supplicant using a KEK and distributes both the PTK and GTK to the AP over the internal trusted channel protected by DTLS. The GTK is also protected with an AES Key Wrap. The GTK is used to protect multicast/broadcast traffic and is shared among all Supplicants and the AP."

5.1.3.2 Guidance Documentation

574 There are no Guidance assurance activities.

5.1.3.3 Tests

Technical Decision: This test was modified per TD0315.

575 The evaluator shall also perform the following test using a packet sniffing tool to collect frames between a wireless client and the TOE (which may be performed in conjunction with the assurance activity for FCS_CKM.1.1(2).

576 To fully test the broadcast/multicast functionality, these steps shall be performed as the evaluator connects multiple clients to the TOE. The evaluator shall ensure that GTKs established are sent to the appropriate participating clients.

577 Step 1: The evaluator shall configure the access point to an unused channel and configure the WLAN sniffer to sniff only on that channel (i.e., lock the sniffer on the selected channel). The sniffer should also be configured to filter on the MAC address of the TOE and/or client.

578 Step 2: The evaluator shall configure the TOE to communicate with the client using IEEE 802.11-2012 and a 256-bit (64 hex values 0-f) pre-shared key, setting up the connections as described in the operational guidance. The pre-shared key is only used for testing.

579 Step 3: The evaluator shall start the sniffing tool, initiate a connection between the TOE and client, and allow the client to authenticate, associate and successfully complete the 4-way handshake with the TOE.

580 Step 4: The evaluator shall set a timer for 1 minute, at the end of which the evaluator shall disconnect the TOE from the client and stop the sniffer.

581 Step 5: The evaluator shall identify the 4-way handshake frames (denoted EAPOL-key in Wireshark captures) and derive the PTK and GTK from the 4-way handshake frames and pre- shared key as specified in IEEE 802.11-2012.

582 Step 6: The evaluator shall select the first data frame from the captured packets that was sent between the TOE and client after the 4-way handshake successfully completed, and with the frame control value 0x4208 (the first 2 bytes are 08 42). The evaluator shall use the GTK to decrypt the data portion of the selected packet as specified in IEEE 802.11-2012, and shall verify that the decrypted data contains ASCII-readable text.

583 Step 7: The evaluator shall repeat Step 6 for the next 2 data frames with frame control value 0x4208.

High-Level Test Description
Sniff the wireless traffic between the AP and two wireless clients and initiate a connection to the test WLAN for each client using PSK authentication. Examine the captured wireless frames and verify all 4 EAPOL handshake messages are captured for at least one client, and that each client's authentications are successful. Let the capture run for 1 minute after authentication takes place and send broadcast traffic through the wireless channel to generate broadcast data frames. Use wireshark's built in IEEE802.11 WPA decryption functionality to decrypt 3 data frames using the GTK. Verify the data contains ASCII readable text.
Findings: PASS

Technical Decision: These tests were added per TD0282.

AES Key Wrap (AES-KW) Tests

584 Test 1: The evaluator shall test the authenticated encryption functionality of AES-KW for EACH combination of the following input parameter lengths:

128 and 256 bit key encryption keys (KEKs)

Three plaintext lengths. One of the plaintext lengths shall be two semi-blocks (128 bits). One of the plaintext lengths shall be three semi-blocks (192 bits). The third data unit length shall 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 shall use the AES-KW authenticated-encryption function of a known good implementation.

Findings: See Table 24 in the [ST] for Wifi Alliance certificates.

AES Key Wrap (KW) (as defined in NIST SP 800-38F) See Test 2 below.

585 Test 2: The evaluator shall 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. Additionally, the evaluator shall modify one byte of the ciphertext, attempt to decrypt the modified ciphertext, and ensure that a failure is returned rather than plaintext.

Findings: See Table 24 in the [ST] for Wifi Alliance certificates.

AES Key Wrap (KW) (as defined in NIST SP 800-38F)

TOE Component	**Cryptographic operation**	**NIST Standard**	**SFR(s) supported**	**CAVP algorithm list name (e.g. AES, KAS, CVE, etc.)**	**CAVP certificate number**
IW6300 ESW6300 AP 1562 AP 2802 AP 3802 AP 4800	AES-KW	AES Key Wrap (KW) (as defined in NIST SP 800-38F)	FCS_CKM.2/GTK	AES-KW	A2452
Catalyst 9130 Catalyst 9115 Catalyst 9120 Catalyst 9105	AES-KW	AES Key Wrap (KW) (as defined in NIST SP 800-38F)	FCS_CKM.2/GTK	AES-KW	A877
Catalyst 9800-80 Catalyst 9800-40 Catalyst 9800-L Catalyst 9800-CL	AES-KW	AES Key Wrap (KW) (as defined in NIST SP 800-38F)	FCS_CKM.2/GTK	AES-KW	A2452 A1462

CAVP A2452 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=14941

CAVP A877 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13370

CAVP A1462 - https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/details?product=13937

AES Key Wrap with Padding (AES-KWP) Tests

586 Test 1: The evaluator shall test the authenticated-encryption functionality of AES-KWP for EACH combination of the following input parameter lengths:

128 and 256 bit key encryption keys (KEKs)

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

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

Findings: AES-KWP functionality is not claimed by the TOE.

587 Test 2: The evaluator shall 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. Additionally, the evaluator shall modify one byte of the ciphertext, attempt to decrypt the modified ciphertext, and ensure that a failure is returned rather than plaintext.

Findings: AES-KWP functionality is not claimed by the TOE.

5.1.4 FCS_CKM.2/PTK – FCS_CKM.2(4) Cryptographic Key Distribution

5.1.4.1 TSS

588 The evaluator shall examine the TSS to determine that it describes which keys are distributed outside the TOE, where they are sent, and the purpose for this transfer.

Findings: [ST] / TOE Summary Specification states, “The Authenticator securely distributes the GTK to the Supplicant using a KEK and distributes both the PTK and GTK to the AP over the internal trusted channel protected by DTLS. “

“The GTK is used to protect multicast/broadcast traffic and is shared among all Supplicants and the AP. The Pairwise Transient Key (PTK) is used to protect unicast traffic with a single Supplicant.”

Application note from ST states: “This requirement refers to the PTK derived by the WLC (Authenticator) and distributed to the AP.”

5.1.4.2 Guidance Documentation

589 If this is dependent on configuration of the System, the evaluator shall confirm that the operational guidance contains instructions for how to configure that the keys are adequately protected.

Findings:	The DTLS-CAPWAP, CC Mode and Enable LSC Provisioning for AP subsections of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] describe the necessary configurations required to ensure keys are adequately protected during distribution. The [AGD] does not identify any further configurations required to ensure keys are adequately protected.

5.1.4.3 Tests

590 This requirement will be tested in conjunction with the tests for the cryptographic primitives, the secure protocols, and FPT_ITT.

5.1.5 FCS_RADSEC_EXT.1 RADIUS over TLS

TD0271: RADsec as alternative to IPsec

5.1.5.1 TSS

591 The evaluator shall verify that the TSS description includes the use of RADIUS over TLS, as described in RFC 6614.

592 If X.509v3 certificates is selected, the evaluator shall ensure that the TSS description includes the use of client-side certificates for TLS mutual authentication.

Findings:	[ST] / TOE Summary Specification states, “The TSF implements RFC 6614 to provide secure TLS communication between itself and an external RADIUS server (RADsec).” [ST] / TOE Summary Specification for FCS_TLSC_EXT.2 states, “The TOE supports TLS mutual authentication and will present a client certificate to the RADsec server and EST Server during connection establishment.”

5.1.5.2 Guidance Documentation

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

Findings:	The TLS-RADsec subsection of the section, Preparative Procedures and Operational Guidance for the TOE in the [AGD] provides instructions on how to configure RADsec on the TOE so that it meets the requitement. The description includes instructions on configuring X.509v3 certificates for TLS mutual authentication. The subsection states, “RADIUS over TLS (RADsec) is used by the Controller to securely access the RADIUS server. The steps below provide instructions to configure RADIUS over TLS. Since TLS mutual authentication is required, you will need to generate a private key and enrol the intermediate trustpoint for a certificate. Radius TLS supports the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite.”

5.1.5.3 Tests

594 The evaluator shall demonstrate the ability to successfully establish a RADIUS over TLS connection with a RADIUS server. This test shall be performed with X.509v3 certificates if selected and performed with pre-shared keys if selected.

Findings:	This test is covered by FCS_TLSC_EXT.1 Extended (RADsec) in the NDcPP Test Plan/Findings document.

5.2 Identification and Authentication (FIA)

5.2.1 FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition

5.2.1.1 TSS

595 The evaluator shall examine the TSS to ensure that it identifies all protocols that allow both text-based and bit-based pre-shared keys, and states that text-based pre-shared keys of 22 characters are supported. For each protocol identified by the requirement, the evaluator shall confirm that the TSS states the conditioning that takes place to transform the text-based pre-shared key from the key sequence entered by the user (e.g., ASCII representation) to the bit string used by the protocol, and that this conditioning is consistent with the last selection in the FIA_PSK_EXT.1.3 requirement.

596 The evaluator shall also examine the TSS to ensure it describes the process by which the bit-based pre-shared keys are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG_EXT.1.

Findings:	[ST] / TOE Summary Specification states, “The TOE supports use of pre-shared keys for authentication of IPsec peers between the WLC component and a remote syslog server. Pre-shared keys can be entered as ASCII characters and must be 22 characters long. Pre-shared keys can also be entered as HEX (“bit-based”) values.” FIA_PSK_EXT does not contain description of conditioning that takes place to transform the text-based pre-shared key to bit string. See below for rationale from the vendor: “There cannot be an assurance activity for a non-existent SFR.” “There is no pre-shared key conditioning SFR element in [WLAN EP].” “Some older NIAP PPs contained a conditioning SFR such as:” “FIA_PSK_EXT.1.3 The TSF shall condition the text-based pre-shared keys by using [selection: SHA-1, SHA-256, SHA-512, [assignment: method of conditioning text string]].” “However the WLAN EP removed the conditioning SFR element prior to its publication in 2015. The reason it was removed is unless both IPsec peers happen to support exactly the same type of conditioning, there will be interoperability issues.” “The PP author must have erroneously left this TSS Assurance Activity in.”

5.2.1.2 Guidance Documentation

597 The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on the composition of strong text-based pre-shared keys, and (if the selection indicates keys of various lengths can be entered) that it provides information on the range of lengths supported. The guidance must specify the

allowable characters for pre-shared keys, and that list must be a super-set of the list contained in FIA_PSK_EXT.1.2.

598 The evaluator shall confirm the operational guidance contains instructions for either entering bit-based pre-shared keys for each protocol identified in the requirement, or generating a bit-based pre-shared key (or both).

Findings:	The TOE uses pre-shared keys for IPsec. The IPsec subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD] provides guidance to administrators on composition of strong text-based pre-shared keys including length requirements and supported characters. The evaluator confirmed the supported characters are the same as those contained in FIA_PSK_EXT.1.2. The guidance provides instructions on entering a bit-based pre-shared keys for IPsec in a hexadecimal format. The subsection states, “e. Specify a pre-shared key. To specify a text-based pre-shared key: WLC(config-ikev2-keyring-peer)# pre-shared-key 0 <pre-shared key> Note: Pre-shared keys on the TOE must be at least 22 characters in length and can be composed of any combination of upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”). To specify a bit-based pre-shared key: WLC(config-ikev2-keyring-peer)# pre-shared-key hex <pre-shared key in hex>”

5.2.1.3 Tests

599 The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE). Note that one or more of these tests can be performed with a single test case.

600 Test 1: The evaluator shall compose a pre-shared key of 22 characters that contains a combination of the allowed characters in accordance with the operational guidance, and demonstrates that a successful protocol negotiation can be performed with the key.

High-Level Test Description
Initiate an IPsec connection from the WLC to the test workstation using PSK authentication with a password that is 22 characters containing a combination of allowed characters in accordance with the operational guidance. Verify the connection succeeds using PSK authentication by inspection of the audit logs and associated traffic capture.
Findings: PASS

601 Test 2 [conditional]: If the TOE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length; the maximum length; a length inside the allowable range; and invalid lengths beyond the supported range (both higher and lower). The minimum, maximum, and included length tests should be successful, and the invalid lengths must be rejected by the TOE.

Findings:	Multiple pre-shared key lengths are not claimed in the [ST].

602 Test 3 [conditional]: If the TOE does not generate bit-based pre-shared keys, the evaluator shall obtain a bit-based pre-shared key of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.

High-Level Test Description
Initiate an IPsec connection from the WLC to the test workstation using PSK authentication with a password that is 22 characters containing a combination of allowed characters in accordance with the operational guidance. Ensure a bit-based PSK is used to build the TOE IPsec configuration. Verify the connection succeeds using PSK authentication by inspection of the audit logs and associated traffic capture.
Findings: PASS

603 Test 4 [conditional]: If the TOE does generate bit-based pre-shared keys, the evaluator shall generate a bit-based pre-shared key of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.

Findings:	The ST does not claim support for bit-based key generation.

5.2.2 FIA_UAU.6 Re-authenticating

5.2.2.1 TSS

604 There are no TSS assurance activities.

5.2.2.2 Guidance Documentation

605 There are no Guidance assurance activities.

5.2.2.3 Tests

606 The evaluator shall perform the following test for each of the conditions specified in the requirement:

607 Test 1: The evaluator shall attempt to change their password as directed by the operational guidance. While making this attempt, the evaluator shall verify that re-authentication is required.

High-Level Test Description
As directed by the operational guidance, attempt to change the password for the administrative user. While making the attempt, verify that re-authentication as the administrative user is required.
Findings: PASS

5.2.3 FIA_8021X_EXT.1 Extended: 802.1X Port Access Entity (Authenticator) Authentication

5.2.3.1 TSS

608 In order to show that the TSF implements the 802.1X-2010 standard correctly, the evaluator shall ensure that the TSS contains the following information:

- The sections (clauses) of the standard that the TOE implements;
- For each identified section, any options selected in the implementation allowed by the standards are specified; and
- For each identified section, any non-conformance is identified and described, including a justification for the non-conformance.

609 Because the connection to the RADIUS server will be contained in an IPsec tunnel (FCS_IPSEC_EXT.1), the security mechanisms detailed in the RFCs identified in the requirement are not relied on to provide protection for these communications. Consequently, no extensive analysis of the RFCs is required. However, the evaluator shall ensure that the TSS describes the measures (documentation, testing) that are taken by the product developer to ensure that the TOE conforms to the RFCs listed in this requirement.

Findings:	[ST] / TOE Summary Specification states, "The TSF strictly follows port-based network control as defined in Clause 7.1 and EAP as defined in Clause 8 and Clause 11 of [IEEE 802.1X-2010]." "The TSF implements PRF-384 and PRF-704 key derivation algorithms as specified in [IEEE 802.11-2012] and [IEEE 802.11ac-2013] respectively, to derive the number of bits required to obtain Pairwise Transient Key (PTK) and Group Temporal Key (GTK) keys." "Certification testing performed by the Wi-Fi Alliance demonstrates the TOE implements the IEEE 802.11-2012 standard correctly. Refer to Table 24 for identification of the relevant Wi-Fi Alliance certificates."

5.2.3.2 Guidance Documentation

610 There are no guidance assurance activities.

5.2.3.3 Tests

611 Test 1: The evaluator shall demonstrate that a wireless client has no access to the test network. After successfully authenticating with a RADIUS server through the TOE, the evaluator shall demonstrate that the wireless client does have access to the test network.

High-Level Test Description
Confirm network access is unavailable by pinging a host on the test network without prior authentication. Initiate a client connection to the test WLAN using 802.1X authentication. Examine the WLC logs and wireless client console output and verify the client authentication is successful.

High-Level Test Description
Confirm network access is available by pinging a host on the test network after successful authentication.
Findings: PASS

612 Test 2: The evaluator shall demonstrate that a wireless client has no access to the test network. The evaluator shall attempt to authenticate using an invalid client certificate, such that the EAP-TLS negotiation fails. This should result in the wireless client still being unable to access the test network.

High-Level Test Description
Confirm network access is unavailable by pinging a host on the test network without prior authentication. Initiate a client connection to the test WLAN using 802.1X authentication with a bad client certificate. Examine the WLC logs and wireless client console output and verify the client authentication is unsuccessful. Confirm network access is still unavailable by pinging a host on the test network after the authentication attempt.
Findings: PASS

613 Test 3: The evaluator shall demonstrate that a wireless client has no access to the test network. The evaluator shall attempt to authenticate using an invalid RADIUS certificate, such that the EAP-TLS negotiation fails. This should result in the wireless client still being unable to access the test network.

614 Note: Tests 2 and 3 above are not tests that "EAP-TLS works", although that's a by-product of the test. The test is actually that a failed authentication (under two failure modes) results in denial of access to the network, which is the 3rd element of this component.

High-Level Test Description
Confirm network access is unavailable by pinging a host on the test network without prior authentication. Initiate a client connection to the test WLAN using 802.1X authentication with a bad EAP-TLS RADIUS server certificate configured on the RADIUS server. Examine the WLC logs and wireless client console output and verify the client authentication is unsuccessful. Confirm network access is still unavailable by pinging a host on the test network after the authentication attempt.
Findings: PASS

5.3 Security Management (FMT)

5.3.1 FMT_SMR.1 Security Management Roles

5.3.1.1 TSS

615 There are no TSS assurance activities.

5.3.1.2 Guidance Documentation

616 The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration.

Findings:	The [AGD] provides instructions for administering the TOE locally and remotely. Necessary configurations for administering the TOE remotely over SSH and HTTPS are provided in the Remote Administration Protocols subsection of the section, Preparative Procedures and Operational Guidance for the TOE of the [AGD]. The [AGD] does not describe any necessary client configuration for remote administration. The subsection states, "SSH is used to securely access the CLI from a remote workstation. The steps below provide instructions to configure SSH Server for the CC evaluated configuration. For additional information on SSH refer to the "Configuring Secure Shell" Chapter of [12]." and "HTTPS is used by the Administrator to securely access the WebGUI from a remote workstation. The steps below provide instructions to configure HTTPS. For additional information on HTTPS refer to the "Configuring Secure Socket Layer HTTP" Chapter of [12]."

5.3.1.3 Tests

617 The evaluator shall perform the following test:

618 Test 1: The evaluator shall demonstrate that after configuring the TOE for first use from the operational guidance, it is possible to establish an administrative session with the TOE on the "wired" portion of the device. They shall then demonstrate that an identically configured wireless client that can successfully connect to the TOE cannot be used to perform administration.

High-Level Test Description
Attempt to establish an administrative session using SSH and the Web GUI with the WLC through the wired interface of the host. Verify the attempts succeed. Attempt to establish an administrative session using SSH and the Web GUI, from the wireless interface of the host. Verify such attempts fail. Ping the administrative interface of the WLC from the wireless interface of the host. Verify the ping succeeds.
Findings: PASS

5.4 Protection of the TSF (FPT)

5.4.1 FPT_TST_EXT.1 Extended: TSF Testing

619 The evaluator shall perform the following activities in addition to the assurance activity specified in the base NDcPP for this SFR:

5.4.1.1 TSS

620 The evaluator shall examine the TSS to ensure that it details the self tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

Findings: [ST] / TOE Summary Specification details the self-tests and what the tests are doing.

Below is an example of one of the descriptions of one of the self-tests that are performed:

"■ HMAC Known Answer Test - For each of the hash values listed, the HMAC implementation is fed known plaintext data and a known key. These values are used to generate a MAC. This MAC is compared to a known MAC to verify that the HMAC and hash operations are operating correctly."

"All TOE components (WLC and AP) will automatically verify the integrity of the stored image when loaded for execution."

"These tests are sufficient to verify correct operation of cryptographic modules."

621 The evaluator shall examine the TSS to ensure that it describes how to verify the integrity of stored TSF executable code when it is loaded for execution, which includes the generation and protection of the "check value" used to ensure integrity as well as the verification step. This description shall also cover the digital signature service used in performing these functions. The evaluator also checks the operational guidance to ensure that any actions required by the administrator to initialize or operate this functionality are present.

Findings: [AGD] / TOE Summary Specification provides the following:
"All TOE components (WLC and AP) will automatically verify the integrity of the stored image when loaded for execution.

The WLC uses a Cisco public key to validate the digital signature to obtain an embedded SHA512 hash that was generated prior to the image being distributed from Cisco. The WLC then computes its own hash of the image using the same SHA512 algorithm. The WLC verifies the computed hash against the embedded hash. If they match the image is authenticated and has not been modified or tampered. If they do not match the image will not boot or execute.

All hardware WLC appliances will display at bootup a message that the image was successfully validated:

"RSA Signed RELEASE Image Signature Verification Successful."

After boot, the authorized administrator can also manually verify the digital signature by executing on the WLC:

verify bootflash:<image or package name>

	The AP will perform a digital signature verification check on its stored image. When successfully validated the AP will display at bootup: "Image signing verification success, continue to run…" If integrity of the stored image is not successfully verified the image will not boot or execute.”

5.4.1.2 Guidance

622 The evaluator also ensures that the TSS (or the operational guidance) describes the actions that take place for successful (e.g. hash verified) and unsuccessful (e.g., hash not verified) cases.

Findings:	[ST] / TOE Summary Specification provides examples of successful integrity checking messages for the WLC and AP and provides descriptions of unsuccessful cases. [AGD] / Auditing provides examples of unsuccessful TSF self-tests that are run on boot.

5.4.1.3 Tests

623 The evaluator shall perform the following tests:

624 Test 1: Following the operational guidance, the evaluator shall initialize the integrity protection system. The evaluator shall perform actions to cause TSF software to load and observe that the integrity mechanism does not flag any executables as containing integrity errors.

Findings:	Integrity tests are performed on boot, and examples of successful integrity checking are performed as part of FPT_TST_EXT.1 testing done in NDcPP Test Plan and Findings documents.

625 Test 2: The evaluator shall modify the TSF executable, and cause that executable to be loaded by the TSF. The evaluator shall observe that an integrity violation is triggered (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 the module was modified so that it was rendered unable to run because its format was corrupt).

Findings:	Testing of a modified TSF executable (binary) were performed in NDcPP FPT_TUD_EXT.1 Test 2.

5.4.2 FPT_FLS.1 Failure with preservation of secure state

5.4.2.1 TSS

626 The evaluator shall review the TSS section to determine that the TOE's implementation of the fail secure functionality is documented. The evaluator shall first examine the TSS section to ensure that all failure modes specified in the ST are described. The evaluator shall review the TSS to determine that the definition of secure state is defined and is suitable to ensure protection of key material and user data.

Findings:	[ST] / TOE Summary Specification states, "If a critical failure occurs that results in the TOE ceasing operation, the TOE securely disables its interfaces to prevent the unintentional flow of any information and then will reload. If the failure persists the TOE will continue to reload. This functionally prevents any failure from causing an unauthorized information flow. There are no failures that circumvent this protection." The evaluator determined that this is suitable to ensure protection of key material and user data.

5.4.2.2 Guidance Documentation

627 There are no assurance activities.

5.4.2.3 Tests

628 For each failure mode specified in the ST, the evaluator shall ensure that the TOE attains a secure state (shutdown) after initiating each failure mode type.

High-Level Test Description
Using vendor provided evidence, verify the POST is performed by the TOE on start-up and that the TOE attains a secure state if the POST fails, as described in the TSS.
Findings: PASS

5.5 TOE Access (FTA)

5.5.1 FTA_TSE.1 TOE Session Establishment

5.5.1.1 TSS

629 The evaluator shall examine the TSS to determine that all of the attributes on which a client session can be denied are specifically defined.

Findings:	[ST] / TOE Summary Specification states, "The Administrator can deny establishment of wireless client sessions based on SSID, time, day attributes. The SSID is the name for a wireless network defined by the Security Administrator. To deny based on time or day attributes, the Administrator from the WLC defines "calendar profile" and tags that to the "wireless profile policy". The wireless clients where the Administrator has applied the "wireless profile policy" are denied access to WLAN during the configured day and/or time."

5.5.1.2 Guidance Documentation

630 The evaluator shall examine the operational guidance to determine that it contains guidance for configuring each of the attributes identified in the TSS.

Findings:	The evaluator confirmed the [AGD] provides guidance for configuring all attributes identified in the TSS.

5.5.1.3 Tests

631 The evaluator shall also perform the following test for each attribute:

632 Test 1: The evaluator successfully establishes a client session with a wireless client. The evaluator then follows the operational guidance to configure the system so that that client's access is denied based on a specific value of the attribute. The evaluator shall then attempt to establish a session in contravention to the attribute setting (for instance, the client is denied WLAN access based upon the TOE interface (e.g. WLAN access point) it is connecting to or the client is denied access based upon the time-of-day or day-of-week it is attempting connection on). The evaluator shall observe that the access attempt fails.

High-Level Test Description
Configure the system to deny a client's access to the WLAN based on the TOE interface (SSID). Verify the client's attempt to access the restricted WLAN is denied and an appropriate audit message appears. Note that this portion of the test is implicitly satisfied due to the way Calendar Profiles are applied. Calendar Profiles are specified within a Policy Profile which is associated with a specific WLAN/SSID. Configure the system to deny a client's access to the WLAN based on the day. Verify the client's attempt to access the restricted WLAN is denied and an appropriate audit message appears. Configure the system to deny a client's access to the WLAN based on the time. Verify the client's attempt to access the restricted WLAN is denied and an appropriate audit message appears. Configure the system to deny a client's access to the WLAN based on the day, for all days except the current day. Verify the client's attempt to access the restricted WLAN is accepted. Configure the system to deny a client's access to the WLAN based on the time, for all times, except the current time. Verify the client's attempt to access the restricted WLAN is accepted.
Findings: PASS

6 Evaluation Activities for Security Assurance Requirements

6.1 ASE: Security Target

633 When evaluating a Security Target, the evaluator performs the work units as presented in the CEM. In addition, the evaluator ensures the content of the TSS in the ST satisfies the EAs specified in Section 2 (Evaluation Activities for SFRs).

Findings:	See above sections.

634 For distributed TOEs only the SFRs classified as 'all' have to be fulfilled by all TOE parts. The SFRs classified as 'One' or 'Feature Dependent' only have to be fulfilled by either one or some TOE parts, respectively. To make sure that the distributed TOE as a whole fulfills all the SFRs the following actions for ASE_TSS.1 have to be performed as part of ASE_TSS.1.1E.

ASE_TSS.1 element	Evaluator Action
ASE_TSS.1.1C	The evaluator shall examine the TSS to determine that it is clear which TOE components contribute to each SFR or how the components combine to meet each SFR. The evaluator shall verify the sufficiency to fulfil the related SFRs. This includes checking that the TOE as a whole fully covers all SFRs and that all functionality that is required to be audited is in fact audited regardless of the component that carries it out.

Findings:	See above sections.

6.2 ADV: Development

635 The design information about the TOE is contained in the guidance documentation available to the end user as well as the TSS portion of the ST, and any required supplementary information required by this cPP that is not to be made public.

636 The functional specification describes the TOE Security Functions Interfaces (TSFIs). It is not necessary to have a formal or complete specification of these interfaces.

637 No additional "functional specification" documentation is necessary to satisfy the Evaluation Activities specified in [SD].

638 The Evaluation Activities in [SD] are associated with the applicable SFRs; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.

639 5.2.1.1 Evaluation Activity: The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

Findings:	From section 7.2.1 of the NDcPP : "For this cPP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation." The [ST] and the AGD comprise the functional specification. If the test in [SD] cannot be completed because the [ST] or the AGD is incomplete, then the functional specification is not complete and observations are required. During the evaluator's use of the product and its interfaces (the Web GUI, SSH CLI, local serial port), there were no areas that were deficient.

640 5.2.1.2 Evaluation Activity: The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

Findings:	See comments in the previous work unit.

641 5.2.1.3 Evaluation Activity: The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

Findings:	See comments in the previous work unit.

6.3 AGD: Guidance

642 The design information about the TOE is contained in the guidance documentation available to the end user as well as the TSS portion of the ST, and any required supplementary information required by this cPP that is not to be made public.

643 5.3.1.1 Evaluation Activity: The evaluator shall ensure the Operational guidance documentation is distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that Security Administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

Findings:	[AGD] section "Obtaining Documentation and Submitting a Service Request" provides a link for administrators and users to obtain a notification when any new Cisco product attains CC certification and a link to its operational guidance documentation. https://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html

644 5.3.1.2 Evaluation Activity: The evaluator shall ensure that the Operational guidance is provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

Findings:	There is only one operational environment claimed in the [ST]. All TOE platforms claimed in [ST] are covered by the operational guidance. This is evidenced by the platform equivalency.

645 5.3.1.3 Evaluation Activity: The evaluator shall ensure that the Operational guidance contains instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

Findings:	[AGD] provides instructions on configuring the FIPS Mode and verifying FIPS mode is enabled.

646 5.3.1.4 Evaluation Activity: The evaluator shall ensure the Operational guidance makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

Findings:	[AGD] specifies all the interfaces and protocols used by the TOE in the evaluated configuration. The [AGD] also lists the Excluded Functionality. The Operational guidance makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

5.3.1.5 Evaluation Activity

647 In addition the evaluator shall ensure that the following requirements are also met.

a) The guidance documentation shall contain instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

Note: Updated per TD0536.

b) The documentation must describe the process for verifying updates to the TOE for each method selected for FPT_TUD_EXT.1.3 in the Security Target. The evaluator shall verify that this process includes the following steps:

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

6) Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes instructions that describe at least one method of validating the hash/digital signature.

c) The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The guidance documentation shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.

Findings:	See work unit [PP] 5.3.1.3 for configuration of the cryptographic engine. [AGD] sections “Verify TOE software” and “Upgrade TOE Software” provide instructions for download and verification of the TOE updates. See work unit [PP] 5.3.1.4 for details as to what was covered by the EAs.

648 5.3.2.1 Evaluation Activity: The evaluator shall examine the Preparative procedures to ensure they include a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target).

Findings:	Please refer to work unit AGD_OPE.1-6.

649 5.3.2.2 Evaluation Activity: The evaluator shall examine the Preparative procedures to ensure they are provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

Findings:	[AGD] section “Procedures and Operational Guidance for IT Environment” provides instructions for configuration of the Operational Environment. The evaluator verified that this addresses all claimed platforms.

650 5.3.2.3 Evaluation Activity: The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment.

Findings:	[AGD] section “Preparative Procedures and Operational Guidance for the TOE” provides instructions for the secure installation of the TOE. This covers all claimed Operational Environments.

651 5.3.2.4 Evaluation Activity: The evaluator shall examine the preparative procedures to ensure they include instructions to manage the security of the TSF as a product and as a component of the larger operational environment.

Findings:	The guidance documentation provides extensive information on managing the security of the TOE as an individual product. Additional best practice guidance provided within those documents helps instil a culture of secure manageability within a larger operational environment.

652 In addition the evaluator shall ensure that the following requirements are also met.

The preparative procedures must:

a) include instructions to provide a protected administrative capability; and

b) identify TOE passwords that have default values associated with them and instructions shall be provided for how these can be changed.

Findings:	[AGD] section “Preparative Procedures and Operational Guidance for the TOE” specifies the secure installation of the TOE to provide a protected interface. There are no default passwords identified in the TOE. In the [AGD] section “Preparative Procedures and Operational Guidance for the TOE”, when the TOE is initially booted up it enters the initial configuration mode, and the administrator must configure a new administrator password.

7 Vulnerability Assessment

NOTE:	Modified per TD0547.

653 5.6.1.1 Evaluation Activity: The evaluator shall examine the documentation outlined below provided by the developer to confirm that it contains all required information. This documentation is in addition to the documentation already required to be supplied in response to the EAs listed previously.

654 The developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components should identify at a minimum the processors used by the TOE. Software components include applications, the operating system and other major components that are independently identifiable and reusable (outside of the TOE), for example a web server, protocol or cryptographic libraries, (independently identifiable and reusable components are not limited to the list provided in the example). This additional documentation is merely a list of the name and version number of the components and will be used by the evaluators in formulating vulnerability hypotheses during their analysis.

Findings:	The evaluator collected this information from the developer which was used to feed into the Type 1 Flaw Hypotheses search (below).

655 5.6.1.2 Evaluation Activity: The evaluator formulates hypotheses in accordance with process defined in Appendix A. The evaluator documents the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3. The evaluator shall perform vulnerability analysis in accordance with Appendix A.2. The results of the analysis shall be documented in the report according to Appendix A.3.

Findings:	The following sources of public vulnerabilities were considered in formulating the specific list of flaws to be investigated by the evaluators, as well as to reference in directing the evaluators to perform key-word searches during the evaluation of the TOE. Hypothesis sources for public vulnerabilities were: - Cisco Systems, Inc. security advisories (vendor website) https://tools.cisco.com/security/center/softwarechecker.x - NIST National Vulnerabilities Database (can be used to access CVE and US-CERT databases identified below): https://web.nvd.nist.gov/view/vuln/search - Common Vulnerabilities and Exposures: http://cve.mitre.org/cve/ https://www.cvedetails.com/vulnerability-search.php - US-CERT: http://www.kb.cert.org/vuls/html/search - CISA - Known Exploited Vulnerabilities Catalog: https://www.cisa.gov/known-exploited-vulnerabilities-catalog - OpenSSL Vulnerabilities: https://www.openssl.org/news/vulnerabilities.html - Google

Type 1 Hypothesis searches were conducted on August 30, 2022 and included the following search terms:

- Cisco Catalyst 9800-40 Wireless Controller
- Cisco Catalyst 9800-CL Wireless Controller for Private Cloud (vSphere)
- Cisco Catalyst 9800-80 Wireless Controller
- Cisco Catalyst 9800-L Wireless Controller
- Cisco UCSC-C220-M5
- Cisco UCSC-C240-M5
- Cisco UCSC-C480-M5
- Cisco Catalyst 9130 Series Wi-Fi 6 Access Points
- Cisco Catalyst 9120 Series Wi-Fi 6 Access Points
- Cisco Catalyst 9115 Series Wi-Fi 6 Access Points
- Cisco Catalyst 9105 Series Wi-Fi 6 Access Points
- Cisco Catalyst IW6300 Series Access Points
- Cisco ESW6300 Access Point
- Cisco Aironet 1562 Series Access Points
- Cisco Aironet 4800 Access Point
- Cisco Aironet 3800 Series Access Points
- Cisco Aironet 2800 Series Access Points
- Intel Xeon Silver 4116T
- Intel Xeon Broadwell D-1548
- Intel Xeon Broadwell D-1563N
- Intel Xeon Platinum 8160M
- Qualcomm IPQ8078 ARMv8
- Broadcom BCM49408 ARMv8
- Broadcom BCM47622 ARMv7
- Marvell Armada 390 ARMv8
- ACT2lite (Anti-Counterfeit Technology 2 Lite) 15-14497-02
- Microsemi SmartFusion2 SoC FPGA M2S010TS

Software Components

- Cisco IOS-XE 17.6.01

- OpenResty 1.15.8.3

- CiscoSSL 7.1.3 (based on OpenSSL 1.1.1c)

- IC2M Rel5a

- CiscoSSH 1.7.22 (based on OpenSSH 7.9p1)

- Lightweight AP software 17.6.01

- CiscoSSL 7.1.220 (based on OpenSSL 1.1.1g)

- dnsmasq 2.83-1

- U-Boot 2013 Patch Level 01

- U-Boot 2016 Patch Level 01

- U-Boot 2017 Patch Level 09

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

There is one type-2 hypothesis identified for the NDcPP and is tested in section 4.1 Test 2 of the [AVA] document.

The evaluation team developed Type 3 flaw hypotheses in accordance with Sections A.1.3, A.1.4, and A.2, and no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.

The evaluation team developed Type 4 flaw hypotheses in accordance with Sections A.1.3, A.1.4, and A.2, and no residual vulnerabilities exist that are exploitable by attackers with Basic Attack Potential.

8 Evaluating additional components for a distributed TOE

8.1 Evaluator Actions for Assessing the ST

TSS

656 The evaluator shall examine the TSS to identify any extra instances of TOE components allowed in the ST and shall examine the description of how the additional components maintain the SFRs to confirm that it is consistent with the role that the component plays in the evaluated configuration. For example: the secure channels used by the extra component for intra-TOE communications (FPT_ITT) and external communications (FTP_ITC) must be consistent, the audit information generated by the extra component must be maintained, and the management of the extra component must be consistent with that used for the original instance of the component in the minimum configuration.

Findings:	[ST] / TOE Evaluated Configuration states, “Deployment of the TOE in its evaluated configuration consists of at least one Wireless LAN Controller (WLC) model and at least one Access Point (AP) model specified in table 3.” [ST] / FPT_ITT.1 describes that there are only two types of components in the distributed TOE, the WLC and the AP. All extra instances of TOE components are identified in the [ST] / Table 3 and clearly describes the role the component plays in the evaluated configuration, whether WLC or AP. [ST] / TOE Summary Specification, provides a detailed mapping of SFRs to TOE components as was evaluated as part of ASE_TSS.1.1C of the [SD]. The evaluator determined that the extra instances of the TOE components allowed in the ST maintain the SFRs and are consistent with the role the components play in the evaluated configuration.

8.2 Evaluator Actions for Assessing the Guidance Documentation

Guidance Documentation

657 The evaluator shall examine the description of the extra instances of TOE components in the guidance documentation to confirm that they are consistent with those identified as allowed in the ST. This includes confirmation that the result of applying the guidance documentation to configure the extra component will leave the TOE in a state such that the claims for SFR support in each component are as described in the ST and therefore that all SFRs continue to be met when the extra components are present.

Findings:	[AGD] / Evaluated Configuration describes the extra instances of TOE components and they are consistent with those identified as allowed in the ST.

	[AGD] / Installation provides specific instructions for installing the pND and vND instances of the WLC TOE components. After initial installation there are no differences in configuring the extra instances of WLC components to prepare the TOE securely in the evaluated configuration. There are differences in configuring the extra instances of AP components. The evaluator determined that the result of applying the guidance documentation to configure the extra components will leave the TOE in a state such that all SFRs continue to be met when the extra components are present.

658 The evaluator shall examine the secure communications described for the extra components to confirm that they are the same as described for the components in the minimum configuration (additional connections between allowed extra components and the components in the minimum configuration are allowed of course).

Findings:	There are no differences in configuring the secure communications for the extra components. The evaluator determined that the secure communications are the same as described for the components in the minimum configuration.

8.3 Evaluator Actions for Testing the TOE

Tests

659 The evaluator tests the TOE in the minimum configuration as defined in the ST (and the guidance documentation).

660 If the description of the use of extra components in the ST and guidance documentation identifies any difference in the SFRs allocated to a component, or the scope of the SFRs involved (e.g. if different selections apply to different instances of the component) then the evaluator tests these additional SFR cases that were not included in the minimum configuration.

Findings:	There are no differences between additional components identified in the ST or guidance documentation that impact the SFRs or their scope. The evaluator determined that extra components can be added with no impact to the security function tested in the minimum configuration.

661 In addition, the evaluator tests the following aspects for each extra component that is identified as allowed in the distributed TOE:

• Communications: the evaluator follows the guidance documentation to confirm, by testing, that any additional connections introduced with the extra component and not present in the minimum configuration are consistent with the requirements stated in the ST (e.g. with regard to protocols and ciphersuites used). An example of such an additional connection would be if a single instance of the component is present in the minimum configuration and adding a duplicate component then introduces an extra communication between the two instances. Another example might be if the use of

the additional components necessitated the use of a connection to an external authentication server instead of using locally stored credentials.

• Audit: the evaluator confirms that the audit records from different instances of a component can be distinguished so that it is clear which instance generated the record.

• Management: if the extra component manages other components in the distributed TOE then the evaluator shall follow the guidance documentation to confirm that management via the extra component uses the same roles and role holders for administrators as for the component in the minimum configuration.

Findings:	The evaluator configured each claimed WLC model with a random selection of two AP models, ensuring each unique AP CPU and/or model series was successfully configured with at least one WLC and that the WLC-AP pair had a valid certificate from the Wi-Fi Alliance listed in Table 24 of the ST. The evaluator confirmed that there were no additional connections introduced with the extra components that were not present in the minimum configuration. All communication channels associated with the extra instances are consistent with the requirements stated in the ST, with regard to protocols and ciphersuites used. The evaluator tested each WLC with at least two AP models and confirmed that the audit records from different instances of a component can be distinguished so that it is clear which instance generated the record. Each WLC is explicitly identified in the audit record, with each log entry being prefixed with the IP address of the log source (the WLC). In the evaluated configuration the WLC is the management component. Each distinct extra WLC component was tested, and the evaluator confirmed that the management via the extra components uses the same roles and role holders for administrators as for the component in the minimum configuration.