www.GossamerSec.com

Assurance Activity Report for DataSoft RAP-117

Version 0.3
07/25/23

Prepared by:
Gossamer Security Solutions
Accredited Security Testing Laboratory – Common Criteria Testing
Columbia, MD 21045

Prepared for:
National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

Revision History

Revision	Date	Authors	Summary
Version 0.1	06/14/23	Gossamer	Initial draft
Version 0.2	07/20/23	Gossamer	Updated in response to check out ECR comments
Version 0.3	07/25/23	Gossamer	Addressed ECR comments

The TOE Evaluation was Sponsored by:

DataSoft Corporation
10235 S. 51st Street, Suite 115
Phoenix, AZ 85044

Evaluation Personnel:

- Tyler Catterton
- Tammy Compton

Common Criteria Versions:

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017

Common Evaluation Methodology Versions:

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 5, April 2017

Table of Contents

1. Introduction

This document presents evaluations results of the Datasoft RAP-117 NDcPP22e/WLANAS10/VPNGW12 evaluation. This document contains a description of the assurance activities and associated results as performed by the evaluators.

1.1 Evaluated Platform Equivalence

The Security Target identifies the TOE as the Datasoft RAP-117 (HW version 2.2.0 and FW version 2.2.0). The TOE is a single platform so no equivalence argument is required as the TOE is fully tested.

1.2 CAVP Certificates

The TOE possesses the following cryptographic algorithm certificates:

The TOE's OpenSSL (version 3.1.0 executing on the TOE's i.MX 6UltraLite MCIMX6G2 Processor) library possesses the following cryptographic algorithm certificates:

Requirements	Functions	CAVP Cert
	Cryptographic key generation	
NDcPP22e:FCS_CKM.1	ECC schemes using 'NIST curves' P-256, P-384	A3743
	IKE Peer Auth Cryptographic key generation	
VPNGW12:FCS_CKM.1/IKE	ECC schemes using 'NIST curves' P-384	A3743
	Cryptographic key establishment/distribution	
NDcPP22e:FCS_CKM.2	Elliptic curve-based key establishment schemes: P-256, P-384	A3743
WLANAS10:FCS_CKM.2.1/GTK	AES KW (128 and 256 bits)	A3743
	SSH Encryption/Decryption	
NDcPP22e:FCS_COP.1/DataEncryption	AES GCM (256 bits)	A3743
	IPsec/ESP Encryption/Decryption	
VPNGW12:FCS_COP.1/DataEncryption	AES CBC (256 bits)	A3743
	Cryptographic hashing	
NDcPP22e:FCS_COP.1/Hash	SHA-256/384/512 (digest sizes 256, 384 and 512 bits)	A3743

Requirements	Functions	CAVP Cert
	Keyed-hash message authentication	
NDcPP22e:FCS_COP.1/KeyedHash	HMAC-SHA-384 (key and output MAC size 384)	A3743
	Cryptographic signature services	
NDcPP22e:FCS_COP.1/SigGen	Elliptic Curve Digital Signature Algorithm (ECDSA) with an elliptical curve P-384	A3743
	Random bit generation	
FCS_RBG_EXT.1	CTR_DRBG (AES) with HW based noise sources (256 bits)	A3743

The TOE's Kernel Cryptography (version 5.4 executing on the TOE's i.MX 6UltraLite MCIMX6G2 Processor) library possesses the following cryptographic algorithm certificates:

Requirements	Functions	CAVP Cert
	Encryption/Decryption (802.11 Wi-Fi)	
WLANAS10:FCS_COP.1/DataEncryption	AES CCMP and GCMP (128 and 256 bits)	A3754
	IPsec/ESP Encryption/Decryption	
VPNGW12:FCS_COP.1/DataEncryption	AES GCM (256 bits)	A3754

2. Protection Profile SFR Assurance Activities

This section of the AAR identifies each of the assurance activities included in the claimed Protection Profile and Extended Packages. This section also describes the findings for each activity.

The evidence identified below was used to perform these Assurance Activities.

- DataSoft RAP-117 Security Target, Version 1.3, July 20, 2023 (ST).
- Datasoft RAP-117 WLAN Access System and IPsec VPN Gateway CC Configuration Guide, Version 1.1, July 20, 2023 (AGD)

2.1 Security audit (FAU)

2.1.1 Audit Data Generation (NDcPP22e:FAU_GEN.1)

2.1.1.1 NDcPP22e:FAU_GEN.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.1.1.2 NDcPP22e:FAU_GEN.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

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.

Section 6.1 of the ST states the vendor's Administrative Guidance enumerates the TOE's audit records (generated by the TOE, as the TOE has only a single component and is not a distributed TOE). The TOE identifies cryptographic keys in audit records (records related to the generation/import, changing, or deleting of keys) by their issuer and subject Distinguished Names.

Component Guidance Assurance Activities: 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).

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.

Section 4.0 "Auditable Events" of the Admin Guide includes an example of each required auditable event corresponding with the requirements claimed in the ST and with the associated audit events for those requirements from the NDcPP22e, WLANAS10, and VPNGW12. The evaluator confirmed that this mapping is complete and found examples of each auditable event required by FAU_GEN.1. Additionally, each of the listed events provides the required content matching the FAU_GEN.1 requirement. As part of testing, the evaluator also verified each audit record in the Admin Guide also collected.

From a review of the ST, the Guidance and through testing, the evaluator also determined that the guidance contains all of the administrative actions and their associated audit events that are relevant to the PP and to use of the TOE. These administrative actions are consistent with the security requirements implemented in the TOE and were found to have appropriate management capabilities identified in the guidance documentation.

Component Testing Assurance Activities: 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.

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.

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

The evaluator created a list of the required audit events. The evaluator then collected the audit event when running the other security functional tests described by the protection profiles. For example, the required event for NDcPP22e:FPT_STM.1 is Changes to Time. The evaluator collected these audit records when modifying the clock using administrative commands. The evaluator then recorded these audit events in the proprietary Detailed Test Report (DTR). The security management events are handled in a similar manner. When the administrator was required to set a value for testing, the audit record associated with the administrator action was collected and recorded in the DTR.

2.1.2 Audit Data Generation (VPN Gateway) (VPNGW12:FAU_GEN.1/VPN)

2.1.2.1 VPNGW12:FAU_GEN.1.1/VPN

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.1.2.2 VPNGW12:FAU_GEN.1.2/VPN

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to verify that it describes the audit mechanisms that the TOE uses to generate audit records for VPN gateway behavior. If any audit mechanisms the TSF uses for this are not used to generate audit records for events defined by FAU_GEN.1 in the Base-PP, the evaluator shall ensure that any VPN gateway-specific audit mechanisms also meet the relevant functional claims from the Base-PP.

For example, FAU_STG_EXT.1 requires all audit records to be transmitted to the OE over a trusted channel.

This includes the audit records that are required by FAU_GEN.1/VPN. Therefore, if the TOE has an audit mechanism that is only used for VPN gateway functionality, the evaluator shall ensure that the VPN gateway related audit records meet this requirement, even if the mechanism used to generate these audit records does not apply to any of the auditable events defined in the Base-PP.

Section 6.1 of the ST states the TOE's VPN gateway functionality uses the same underlying audit mechanism (rsyslog) to generate its audits, and thus the TOE transmits VPN audits the same way as it does base-NDcPP audits.

Component Guidance Assurance Activities: The evaluator shall examine the operational guidance to verify that it identifies all security-relevant auditable events claimed in the ST and includes sample records of each event type. If the TOE uses multiple audit mechanisms to generate different sets of records, the evaluator shall verify that the operational guidance identifies the audit records that are associated with each of the mechanisms such that the source of each audit record type is clear.

See NDcPP22e: FAU_GEN.1 where the evaluator confirmed that the Admin Guide identifies all auditable events claimed in the ST and includes sample records.

Component Testing Assurance Activities: The evaluator shall test the audit functionality by performing actions that trigger each of the claimed audit events and verifying that the audit records are accurate and that their format is consistent with what is specified in the operational guidance. The evaluator may generate these audit events as a consequence of performing other tests that would cause these events to be generated.

The audit records associated with VPNGW12:FAU_GEN.1/VPN were collected during testing of that requirement. All required audits including those specified in Table 2 and Table 3 of the PP-Module were found during that testing.

The evaluators further verified that the audits contained the necessary information.

2.1.3 Audit Data Generation (WLANAS10:FAU_GEN.1/WLAN)

2.1.3.1 WLANAS10:FAU_GEN.1.1/WLAN

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: The evaluator will test the TOE's ability to correctly generate audit records by having the TOE generate audit records in accordance with the evaluation activities associated with the functional requirements in this PPModule. When verifying the test results, the evaluator will ensure the audit records generated during testing match the format specified in the administrative guide and that the fields in each audit record have the proper entries.

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

The audit records associated with WLANAS10:FAU_GEN.1/WLAN were collected during testing of that requirement. All required audits including those specified in Table 2 of the PP-Module were found during that testing.

The evaluators further verified that the audits contained the necessary information.

2.1.4 User identity association (NDcPP22e:FAU_GEN.2)

2.1.4.1 NDcPP22e:FAU_GEN.2.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1.

See NDcPP22e:FAU_GEN.1.

Component Guidance Assurance Activities: The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1.

See NDcPP22e:FAU_GEN.1.

Component Testing Assurance Activities: This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

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.

See NDcPP22e:FAU_GEN.1.

2.1.5 Protected Audit Event Storage (NDcPP22e:FAU_STG_EXT.1)

2.1.5.1 NDcPP22e:FAU_STG_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.1.5.2 NDcPP22e:FAU_STG_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.1.5.3 NDcPP22e:FAU_STG_EXT.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

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.

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.

The evaluator shall examine the TSS to ensure that it details the behavior 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.

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 acceptable frequency for the transfer of audit data.

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).

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.

Section 6.1 of the ST states the TOE transmits audit data via syslog transmitted within an IPsec tunnel. The TOE has four gigabytes of internal storage for audit records and should the TOE exhaust this storage space, it overwrites the oldest previous audit records with the new audit records.

The TOE is a standalone TOE that stores audit data locally, and the TOE's logd will rotate logs and once out of space, delete the oldest logs in order to make room for newer logs.

The vendor has designed the TOE to accumulate audit logs while fielded and, upon detecting it has regained network connectivity (typically after returning from the field), to transmit its audit records to the administrator defined remote syslog server. If the TOE can establish an IPsec connection to the admin configured syslog server, it transmits audit records in real-time to the syslog server (while still storing the audit records locally).

Component Guidance Assurance Activities: 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.

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.

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.

Section 2.2 "Syslog Server" of the Admin Guide clarifies the requirements on the audit server.

Any syslog server that can be accessed over IPsec may be used, such as rsyslog partnered with the StrongSwan VPN.

Section 3.13 "Audit Log Configuration" of the Admin Guide states that the TOE can transmit protected (with an IPsec tunnel) audit records to a syslog server in its operational environment. The administrator can use the audit log tab to configure the IP address of the syslog server and set up the VPN to tunnel the syslog data. Once configured, the TOE detects when it has a wired connection (typically post-mission) and attempts to establish an IPsec connection to the administrator specified syslog server and then send audits via syslog.

The default values on this tab will set up a connection over the USB connection to the provisioning PC and are likely sufficient. Update if needed and select "Apply VPN Settings".

In addition to these settings the X.509v3 certificates need to be created and imported into the TOE. This is done via the menu options available when on this tab to Generate New Certificate Request, Import Signed Certificate, Import Trusted CA Certs and Import CRL File.

This setup encrypts all log data as it is exported from the TOE to the provisioning PC. Configuring these properly allows the TOE to do a post-mission export of its log files.

Note that in the TOE's primary use case, it operates without the expectation of network connectivity, and thus the TOE internally accumulates audit logs using its local storage. Upon mission completion, the TOE exports all of its log files. If the TOE has network connectivity to the audit server, the TOE sends protected audit messages to the audit server in real time (but continues to locally store the audit records). The TOE also keeps track of the most recent, successfully transmitted audit record such that if it should lose connectivity with the audit server, it can resume sending records from that point forward upon regaining connectivity.

The TOE does not allow any administrator configurability for audit settings and comes preset with a fixed amount of local storage space and automatically attempted to establish an IPsec protected syslog connection to opportunistically export audit logs.

Component Testing Assurance Activities: 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 test 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.

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).

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.

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.

Test 1: The external audit server was utilizing an rsyslogd 8.16.0. The evaluator observed the TOE initiating an IPsec tunnel to the syslog server and demonstrated that the syslog messages were successfully sent to the syslog server through the encrypted channel. No plaintext traffic was observed in the packet capture. The evaluator also verified that the logs were successfully received by viewing them on the syslog server.

Test 2: The evaluator verified that when the local audit storage was filled, the existing audit data was overwritten using the following rule: Overwrite oldest records first.

Test 3: Not applicable. The TOE does not claim FAU_STG_EXT.2/LocSpace.

Test 4: Not applicable. The TOE is not distributed

2.2 Cryptographic support (FCS)

2.2.1 Cryptographic Key Generation (NDcPP22e:FCS_CKM.1)

2.2.1.1 NDcPP22e:FCS_CKM.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.2 of the ST presents the following table that includes each type of key generated, its usage and its key size.

Purpose	SFR	Scheme	Size
IKE authentication	VPNGW12: FCS_IPSEC_EXT.1.13	ECDSA	P-384
IKE key exchange	VPNGW12: FCS_IPSEC_EXT.1.11	ECDSA	P-256, P-384
WPA3-SAE ECDHE	WLANAS10:FCS_CKM.1/WPA	ECDH	P-256, P-384
SSH authentication	NDcPP22e:FCS_SSHS_EXT.1.5	ECDSA	P-384
SSH key exchange	NDcPP22e:FCS_SSHS_EXT.1.7	ECDH	P-256, P-384

Component Guidance Assurance Activities: 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.

Section 3.8 “Configured Wired, Wi-Fi, and VPN Data” of the Admin Guide states Only AES-GCM-256/IKEv2 DH Group 20 is available in CSfC Compliant mode. The additional modes AES-GCM-256/IKEv2 DH Group 19, AES-CBC-256/IKEv2 DH Group 20 and AES-CBC-256/IKEv2 DH Group 19 are available when the administrator leaves CSfC Compliant mode unchecked.

Section 3.12 “Admin Configuration” of the Admin Guide states The “Add SSH key…” checkbox is available only with CSfC Compliant Mode is unchecked and allows use of 256-bit ECDSA keys for SSH Public Key Authentication.

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.2.

Component Testing Assurance Activities: 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

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

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

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

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.

Key Generation for Elliptic Curve Cryptography (ECC)

FIPS 186-4 ECC Key Generation Test

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

FIPS 186-4 Public Key Verification (PKV) Test

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

Key Generation for Finite-Field Cryptography (FFC)

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

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

- Primes q and p shall both be provable primes

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

and two ways to generate the cryptographic group generator g:

- Generator g constructed through a verifiable process

- Generator g constructed through an unverifiable process.

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

- 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.

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

To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator must seed the TSF parameter generation routine with sufficient data to deterministically generate the parameter set.

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

- g != 0,1

- q divides p-1

- g^q mod p = 1

- g^x mod p = y

for each FFC parameter set and key pair.

FFC Schemes using 'safe-prime' groups

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

(TD0580 applied)

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.2.

2.2.2 Cryptographic Key Generation (for IKE Peer Authentication) - per TD0723 (VPNGW12:FCS_CKM.1/IKE)

2.2.2.1 VPNGW12:FCS_CKM.1.1/IKE

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall check to ensure that the TSS describes how the key-pairs are generated. In order to show that the TSF implementation complies with FIPS PUB 186-4, the evaluator shall ensure that the TSS contains the following information:

- The TSS shall list all sections of Appendix B to which the TOE complies.

- For each applicable section listed in the TSS, for all statements that are not 'shall' (that is, 'shall not', 'should', and 'should not'), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as 'shall not' or 'should not' in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;

- For each applicable section of Appendix B, any omission of functionality related to 'shall' or 'should' statements shall be described;

Any TOE-specific extensions, processing that is not included in the Appendices, or alternative implementations allowed by the Appendices that may impact the security requirements the TOE is to enforce shall be described.

Section 6.2 of the ST presents a table that includes each type of key generated, its usage and its key size. It also states the TOE generates the asymmetric keys listed in the table below. The TOE generated ECDSA key pairs per FIPS 186-4 Appendix B.4.2 (“Testing Candidates”) and does not introduce any TOE-specific extensions, processing or alternative implementations. However, the TOE does diverge from the sole “**should**” found in FIPS 186-4 section B.4.2 (within **Output:** step 2,). The TOE instead internally logs any error encountered during public/private key pair generation and does not output any *d* or *Q* values (in order to fail secure).

Component Guidance Assurance Activities: The evaluator shall check that the operational guidance describes how the key generation functionality is invoked, and describes the inputs and outputs associated with the process for each signature scheme supported. The evaluator shall also check that guidance is provided regarding the format and location of the output of the key generation process.

Section 3.9 “Provisioning the TOE” states that After entering all the configuration data, close the configuration tab select “Provision” in the DPA GUI to push the configuration data to the TOE. With this action the TOE will generate ECDSA P-384 private/public key pairs, generate SAE-PK keys/password, and upload and save a X.509 certificate signing request (CSR) on the provisioning PC. During provisioning, a status bar will show progress and the status line at the bottom of the screen displays current activities being performed. The location of the CSR that needs to be signed by a Certificate Authority (CA) is displayed at the bottom of the UI in the status area.

The above CSR file that was just generated needs to be sent to a CA to be signed. The signed certificate will then be brought back to the Linux PC and imported into the TOE along with the CA certificate chain of the signing CAs.

Component Testing Assurance Activities: For FFC Schemes using 'safe-prime' groups:

Testing for FFC Schemes using safe-prime groups is done as part of testing in FCS_CKM.2.

For all other selections:

The evaluator shall perform the corresponding tests for FCS_CKM.1 specified in the NDcPP SD, based on the selections chosen for this SFR. If IKE key generation is implemented by a different algorithm than the NDcPP key generation function, the evaluator shall ensure this testing is performed using the correct implementation.

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.2.

2.2.3 Cryptographic Key Generation (Symmetric Keys for WPA2 Connections) (WLANAS10:FCS_CKM.1/WPA)

2.2.3.1 WLANAS10:FCS_CKM.1.1/WPA

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The cryptographic primitives will be verified through evaluation activities specified elsewhere in this PPModule. The evaluator will verify that the TSS describes how the primitives defined and implemented by this PP-Module are used by the TOE in establishing and maintaining secure connectivity to the wireless clients. This description will include how the GTK and PTK are generated or derived. The TSS will also provide a description of the developer's methods 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 will 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.

Section 6.2 of the ST states the TOE establishes the GTK and PTK in accordance with IEEE 802.11-2020 and IEEE 802.11ax-2021.

The TOE derives the PTK from the PMK, however, the details of the PMK differs based upon mode. When operating in WPA3-SAE mode, the TOE establishes the PMK from SAE’s dragonfly key exchange.

When operating in WPA3/WPA2-Enterprise mode, the TOE receives the PMK from the Enterprise RADIUS server (the server establishes the PMK from the EAP-TLS exchange).

In both modes, the TOE derives the GTK from the GMK (which the TOE randomly generates) and nonces. The TOE distributes the GTK to wireless clients in an EAPOL-Key frame wrapped using AES Key Wrap in accordance with the IEEE standards.

Upon a successful 4-way handshake, the Authenticator will allow for WLAN data to pass through the system to the Controller in a tunnel architecture or to intended destination in distributed architecture.

The PTK (total 384 bits) is derived into three parts. The second part is KEK and used to encrypt GTK to be sent as 3rd message in WPA2 handshake. Third part is TK, which is actually used to encrypt/decrypt communication between both AP and Client.

The TOE incorporates the NXP (formerly Marvell) 88W8997 Wave 2 Wi-Fi System on Chip (SoC), which NXP subjected to Wi-Fi Alliance testing to demonstrate that the SoC (instantiated in a Test Bed Evaluation Kit) implements the IEEE 802.11-2012 standards correctly. Refer to the Wi-Fi Alliance certificates for compliance, https://www.wi-fi.org/certification.

Device	Wi-Fi Alliance Certificate
Datasoft RAP-117	WFA 72793

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: The evaluator will perform the following test using a packet sniffing tool to collect frames between the TOE and a wireless client:

Step 1: The evaluator will configure the AP 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 client.

Step 2: The evaluator will configure the TOE to communicate with a WLAN client using IEEE 802.11-2020 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.Step 3: The evaluator will start the sniffing tool, initiate a connection between the TOE and WLAN client, and allow the TOE to authenticate, associate, and successfully complete the four-way handshake with the client.

Step 4: The evaluator will set a timer for one minute, at the end of which the evaluator will disconnect the client from the TOE and stop the sniffer.

Step 5: The evaluator will identify the four-way handshake frames (denoted EAPOL-key in Wireshark captures) and derive the PTK from the four-way handshake frames and pre-shared key as specified in IEEE 802.11-2020.

Step 6: The evaluator will select the first data frame from the captured packets that was sent between the client and TOE after the four-way handshake successfully completed and without the frame control value 0x4208 (the first two bytes are 08 42). The evaluator will use the PTK to decrypt the data portion of the packet as specified in IEEE 802.11-2020 and verify that the decrypted data contains ASCII-readable text.

Step 7: The evaluator will repeat Step 6 for the next two data frames between the TOE and client, and without frame control value 0x4208.

Additionally, the evaluator will 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

- Annex J.3 'PRF reference implementation and test vectors' of IEEE 802.11-2020

Step 1: The evaluator configured the access point and configured the sniffer to sniff the 802.11 traffic received by the AP.

Step 2: The evaluator configured the TOE for a 256-bit pre-shared key and setup the connection as described in the operational guidance.

Step 3: The evaluator started Wireshark to capture the 802.11 traffic.

Step 4: The TOE was connected and disconnected after more than one minute while a number of broadcast packets were observed using Wireshark and the capture was collected.

Step 5: The evaluator identified the 4-way handshake frames as well as data frames using Wireshark and collected the packet capture.

Step 6 and 7: After enabling decryption, the evaluator was able to see the frames, albeit properly decrypted as expected.

See Section 1.2 above for the CAVP and Wi-Fi alliance certificates.

2.2.4 Cryptographic Key Establishment (NDcPP22e:FCS_CKM.2)

2.2.4.1 NDcPP22e:FCS_CKM.2.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

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:

Scheme	SFR	Service
RSA	FCS_TLSS_EXT.1	Administration
ECDH	FCS_SSHC_EXT.1	Audit Server
ECDH	FCS_IPSEC_EXT.1	Authentication Server

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.

(TD0580 applied)

Section 6.2 of the ST presents the following table that includes each type of key generated, its usage and its key size.

Purpose	SFR	Scheme	Size
IKE authentication	VPNGW12: FCS_IPSEC_EXT.1.13	ECDSA	P-384
IKE key exchange	VPNGW12: FCS_IPSEC_EXT.1.11	ECDSA	P-256, P-384
WPA3-SAE ECDHE	WLANAS10:FCS_CKM.1/WPA	ECDH	P-256, P-384
SSH authentication	NDcPP22e:FCS_SSHS_EXT.1.5	ECDSA	P-384
SSH key exchange	NDcPP22e:FCS_SSHS_EXT.1.7	ECDH	P-256, P-384

The key establishment schemes in the requirement match those in the NDcPP22e:FCS_CKM.1.1 requirement.

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

Section 3.8 “Configuring Wires, Wi-Fi, and VPN Data” section of the Admin Guide states under the “VPN Section” that Algorithms – List of available encryption algorithms. Only AES-GCM-256/IKEv2 DH Group 20 is available in CSfC

Compliant mode. The additional modes AES-GCM-256/IKEv2 DH Group 19, AES-CBC-256/IKEv2 DH Group 20 and AES-CBC-256/IKEv2 DH Group 19 are available when the administrator leaves CSfC Compliant mode unchecked.

Section 3.12 “Admin Configuration” states, The TOE always supports ecdh-sha2-nistp384 for SSH key exchange, and the administrator can additionally enable support for SSH key exchange using ecdh-sha2-nistp256 by unchecking CSfC mode, and then, from the Admin tab, checking the “Add SSH Key Exchange Algorithm ecdh-sha2-nistp256” checkbox, closing, and applying the configuration.

Component Testing Assurance Activities: Key Establishment Schemes

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

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

Function Test

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

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

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

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

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

Validity Test

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

The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the 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).

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.

RSA-based key establishment

The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses RSAES-PKCS1-v1_5.

FFC Schemes using 'safe-prime' groups

The evaluator shall verify the correctness of the TSF's implementation of safe-prime groups by using a known good implementation for each protocol selected in FTP_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.

(TD0580 applied)

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.2.

2.2.5 Cryptographic Key Distribution (GTK) (WLANAS10:FCS_CKM.2/GTK)

2.2.5.1 WLANAS10:FCS_CKM.2.1/GTK

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator will check the TSS to ensure that it describes how the GTK is wrapped prior to being distributed using the AES implementation specified in this PP-Module, and also how the GTKs are distributed when multiple clients connect to the TOE.

Section 6.2 of the ST states the TOE wraps the GTK with AES Key Wrap in an EAPOL-Key frame and will distribute the GTK after the 4-way handshake when a wireless client first connects as well as after any update to the GTK.

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: The evaluator will 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 evaluation activity for FCS_CKM.1/PMK.

To fully test the broadcast and multicast functionality, these steps will be performed as the evaluator connects multiple clients to the TOE. The evaluator will ensure that GTKs established are sent to the appropriate participating clients.

Step 1: The evaluator will configure the AP 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 client.

Step 2: The evaluator will configure the TOE to communicate with the client using IEEE 802.11-2020 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.

Step 3: The evaluator will start the sniffing tool, initiate a connection between the TOE and client, and allow the client to authenticate, associate and successfully complete the four-way handshake with the TOE.

Step 4: The evaluator will set a timer for one minute, at the end of which the evaluator will disconnect the TOE from the client and stop the sniffer.

Step 5: The evaluator will identify the four-way handshake frames (denoted EAPOL-key in Wireshark captures) and derive the PTK and GTK from the four-way handshake frames and pre-shared key as specified in IEEE 802.11-2020.

Step 6: The evaluator will select the first data frame from the captured packets that was sent between the TOE and client after the four-way handshake successfully completed, and with the frame control value 0x4208 (the first two bytes are 08 42). The evaluator will use the GTK to decrypt the data portion of the selected packet as specified in IEEE 802.11-2020 and verify that the decrypted data contains ASCII-readable text.

Step 7: The evaluator will repeat Step 6 for the next two data frames with frame control value 0x4208.

The evaluator will also perform the following testing based on the supported GTK distribution methods:

AES Key Wrap (AES-KW Tests)

Test 1: The evaluator will 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:

1. One of the plaintext lengths will be two semi-blocks (128 bits).

2. One of the plaintext lengths will be three semi-blocks (192 bits).

3. The third data unit length will be the longest supported plaintext length less than or equal to 64 semi-blocks (4096 bits).

For each combination, generate a set of 100 key and plaintext pairs and obtain the ciphertext that results from AES-KW authenticated encryption. To determine correctness, the evaluator will use the same key and plaintext values and encrypt them using a known good implementation of AES-KW authenticatedencryption, and ensure that the resulting respective ciphertext values are identical.

Test 2: The evaluator will test the authenticated-decryption functionality of AES-KW using the same test as for authenticated-encryption, replacing plaintext values with ciphertext values and AES-KW authenticated-encryption with AES-KW authenticated-decryption. Additionally, the evaluator will modify one byte of the ciphertext, attempt to decrypt the modified ciphertext, and ensure that a failure is returned rather than plaintext.

AES Key Wrap with Padding (AES-KWP Tests)

Test 1: The evaluator will 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 will be one octet. One plaintext length will be 20 octets (160 bits). One plaintext length will be the longest supported plaintext length less than or equal to 512 octets (4096 bits).

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 will use the AES-KWP authenticatedencryption function of a known good implementation.

Test 2: The evaluator will test the authenticated-decryption functionality of AES-KWP using the same test as for AES-KWP authenticated-encryption, replacing plaintext values with ciphertext values and AES-KWP authenticated-encryption with AES-KWP authenticated-decryption. Additionally, the evaluator will modify one byte of the ciphertext, attempt to decrypt the modified ciphertext, and ensure that a failure is returned rather than plaintext.

Step 1-7: See WLANAS10:FCS_CKM.1.1/WPA where a description of testing activities is provided.

AES Key Wrap tests: the TOE has CAVP certificates as identified in Section 1.2.

2.2.6 Cryptographic Key Distribution (PMK) (WLANAS10:FCS_CKM.2/PMK)

2.2.6.1 WLANAS10:FCS_CKM.2.1/PMK

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator will examine the TSS to determine that it describes how the PMK is transferred (that is, through what EAP attribute) to the TOE.

Section 6.2 of the ST states the TOE the TOE receives the PMK in the EAP MS-MPPE-Recv-Key and MS-MPPE-Send-Key RADIUS attributes contained within a RADIUS Access-Accept packet (ultimately transmitted by the Enterprise RADIUS server) within an IPsec protected channel.

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: The evaluator will establish a session between the TOE and a RADIUS server according to the configuration guidance provided. The evaluator will 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.

The evaluator set up a RADIUS server on a Linux machine, which is also capable of capturing packets between the TOE and the RADIUS server. The evaluator also used a mobile device that is capable of reporting the PMK value in the WPA handshake process. The packet capturer was enabled and the mobile device connected so that the evaluator collected both the RADIUS packets for the connection and the PMK. The evaluator then opened up the packet capture in HxD (a hex editor) and searched for the PMK value. The evaluator confirmed that the PMK is not exposed.

2.2.7 Cryptographic Key Destruction (NDcPP22e:FCS_CKM.4)

2.2.7.1 NDcPP22e:FCS_CKM.4.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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 for2). 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.

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).

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.

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 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.

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.

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.

Section 6.2 of the ST presents a table of keys and identifies where each is stored and how it is cleared.

Key	Storage	Encrypted	Destruction and when

	(RAM/Flash)	/Plaintext	
IKE auth keys	RAM	Plaintext	Cleared w/ zeros after use
IKE auth keys	Flash	Plaintext	Stored persistently in the TOE's data partition, and cleared w/ a four-pass overwrite (RNG and zeros) upon zeroization
IKE/ESP SA keys	RAM	Plaintext	Cleared w/ zeros after use
WPA3-SAE ECDH keys	RAM	Plaintext	Cleared w/ zeros after use
WPA2/3 keys	RAM	Plaintext	Cleared w/ zeros after use
SSH host key	RAM	Plaintext	Cleared w/ zeros after use
SSH host key	Flash	Plaintext	Stored persistently in the TOE's data partition, and cleared w/ a four-pass overwrite (RNG and zeros) upon zeroization
SSH session	RAM	Plaintext	Cleared w/ zeros after use

Component Guidance Assurance Activities: 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.

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 [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)] and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

The Admin Gide does not identify any conditions where the key destruction would not occur as described in the TSS.

Component Testing Assurance Activities: None Defined

2.2.8 Cryptographic Operation (AES Data Encryption/Decryption) (NDcPP22e:FCS_COP.1/DataEncryption)

2.2.8.1 NDcPP22e:FCS_COP.1.1/DataEncryption

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE supports AES-256 in the GCM modes in SSH.

Component Guidance Assurance Activities: 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.

Section 3.12 “Admin Configuration” states, the TOE always supports ecdh-sha2-nistp384 for SSH key exchange, and the administrator can additionally enable support for SSH key exchange using ecdh-sha2-nistp256 by unchecking CSfC mode, and then, from the Admin tab, checking the “Add SSH Key Exchange Algorithm ecdh-sha2-nistp256” checkbox, closing, and applying the configuration.

Beyond SSH key exchange, the TOE utilizes a fixed configuration of AES-256-GCM (aes256-gcm@openssh.com) and implicit MAC/integrity algorithms, P-384 ECDSA host keys, and fixed rekey limits of 1 hour and 0.5 Gigabytes (whichever comes first).

Component Testing Assurance Activities: AES-CBC Known Answer Tests

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

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

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

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

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

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

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

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

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

AES-CBC Multi-Block Message Test

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

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

AES-CBC Monte Carlo Tests

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

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

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

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

AES-GCM Test

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.

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

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

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

AES-CTR Known Answer Tests

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. Due to the fact that 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):

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.

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.

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.

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].

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

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

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]

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.

There is no need to test the decryption engine.

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.2.

2.2.9 Cryptographic Operation (AES Data Encryption/Decryption) (VPNGW12:FCS_COP.1/DataEncryption)

2.2.9.1 VPNGW12:FCS_COP.1.1/DataEncryption

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: There is no change to the Evaluation Activities specified for this SFR in the NDcPP Supporting Document. The PP-Module modifies this SFR to require the ST author to make certain selections, but these selections are all part of the original definition of the SFR so no new behavior is defined by the PP-Module.

Section 6.2 of the ST states the TOE supports AES-256 CBC and GCM for all of its IPsec connections (with clients, remote syslog, and RADIUS server). The TOE performs IKE/IPsec AES-GCM using its Kernel Cryptography, while performs AES-CBC using its OpenSSL library.

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.2.10 Cryptographic Operation (AES Data Encryption/Decryption) (WLANAS10:FCS_COP.1/DataEncryption)

2.2.10.1 WLANAS10:FCS_COP.1.1/DataEncryption

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: There are no additional TSS evaluation activities for this component beyond what the NDcPP requires.

Section 6.2 of the ST states the TOE supports AES-CBC in order to comply with FCS_IPSEC_EXT.1 requirement and also supports AES-128/256 CCMP and AES-256 GCMP modes of feedback for 802.11 wireless clients. The TOE encrypts and decrypts Wi-Fi frames using its RAP Kernel Cryptography AES implementation.

Component Guidance Assurance Activities: There are no additional guidance evaluation activities for this component beyond what the NDcPP requires.

Component Testing Assurance Activities: In addition to the tests required by the NDcPP, the evaluator will perform the following testing:

AES-CCM Tests

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

- 128 bit and 256 bit keys

- Two payload lengths. One payload length will be the shortest supported payload length, greater than or equal to zero bytes. The other payload length will be the longest supported payload length, less than or equal to 32 bytes (256 bits).

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

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

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

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 will ensure that these are the only supported lengths. To test the generation-encryption functionality of AES-CCM, the evaluator will perform the following four tests:

Test 1: For each supported key and associated data length and any supported payload, nonce and tag length, the evaluator will supply one key value, one nonce value and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.

Test 2: For each supported key and payload length and any supported associated data, nonce and tag length, the evaluator will supply one key value, one nonce value and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.

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

Test 4: For each supported key and tag length and any supported associated data, payload, and nonce length, the evaluator will supply one key value, one nonce value and 10 pairs of associated data and payload values and, obtain the resulting ciphertext.

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

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

Additionally, the evaluator will use tests from the IEEE 802.11-02/362r6 document 'Proposed Test vectors for IEEE 802.11 TGi', dated September 10, 2002, Section 2.1 AES-CCMP Encapsulation Example and Section 2.2 Additional AES-CCMP Test Vectors to verify further the IEEE 802.11-2020 implementation of AES-CCMP.

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.2.

2.2.11 Cryptographic Operation (Hash Algorithm) (NDcPP22e:FCS_COP.1/Hash)

2.2.11.1 NDcPP22e:FCS_COP.1.1/Hash

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE makes use of hashing within its SSH connections (for remote administration), for the integrity of IPsec/ESP traffic, as part of WPA2/3 PRF functionality, and for signature generation and verification (both IKE peer authentication and trusted updates).

Component Guidance Assurance Activities: The evaluator checks the AGD documents to determine that any configuration that is required to configure the required hash sizes is present.

Section 3.8 “Configuring Wired, Wi-Fi, and VPN Data” of the Admin Guide states The TOE offers no other configuration options beyond the Algorithms and Lifetimes configuration (i.e., the TOE has a fixed configuration for all other IPsec aspects, including hash/HMAC algorithms [HMAC-SHA-384], IKEv2, tunnel mode, and NAT-T support).

Section 3.12 “Admin Configuration” of the Admin Guide states The TOE always supports ecdh-sha2-nistp384 for SSH key exchange, and the administrator can additionally enable support for SSH key exchange using ecdh-sha2-nistp256 by unchecking CSfC mode, and then, from the Admin tab, checking the “Add SSH Key Exchange Algorithm ecdh-sha2-nistp256” checkbox, closing, and applying the configuration.

Beyond SSH key exchange, the TOE utilizes a fixed configuration of AES-256-GCM (aes256-gcm@openssh.com) and implicit MAC/integrity algorithms, P-384 ECDSA host keys, and fixed rekey limits of 1 hour and 0.5 Gigabytes (whichever comes first).

Finally, note that the TOE internally hashes administrator passwords using SHA-512 for protection and does not offer any configurability to use a different hash algorithm

Section 3.15 “RADIUS Configuration” of the Admin Guide states For both its RADIUS and syslog configuration, the TOE offers a fixed configuration supporting, IKEv2, tunnel-mode, NAT-T support, and a fixed set of IKEv2 and ESP algorithms: AES-256-CBC or GCM, SHA-384, and DH Groups 19 and 20 (ECP-256 and 384). The TOE allows no administrator configuration of this fixed configuration other than the SA lifetimes. The TOE allows an administrator to configure the RADIU/Syslog SA lifetime between 3-24 hours and the Child SA Lifetime between 1-8 hours.

Component Testing Assurance Activities: 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.

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

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

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

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

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

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.

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.2.

2.2.12 Cryptographic Operation (Keyed Hash Algorithm) (NDcPP22e:FCS_COP.1/KeyedHash)

2.2.12.1 NDcPP22e:FCS_COP.1.1/KeyedHash

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE uses HMAC keys sized 384 bits with hashes SHA-384 with block size 1024 bits and with output MAC length 384 respectively.

Component Guidance Assurance Activities: 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.

Section 3.8 “Configuring Wired, Wi-Fi, and VPN Data” of the Admin Guide states The TOE offers no other configuration options beyond the Algorithms and Lifetimes configuration (i.e., the TOE has a fixed configuration for all other IPsec aspects, including hash/HMAC algorithms [HMAC-SHA-384], IKEv2, tunnel mode, and NAT-T support).

Section 3.12 “Admin Configuration” of the Admin Guide states The TOE always supports ecdh-sha2-nistp384 for SSH key exchange, and the administrator can additionally enable support for SSH key exchange using ecdh-sha2-nistp256 by unchecking CSfC mode, and then, from the Admin tab, checking the “Add SSH Key Exchange Algorithm ecdh-sha2-nistp256” checkbox, closing, and applying the configuration.

Beyond SSH key exchange, the TOE utilizes a fixed configuration of AES-256-GCM (aes256-gcm@openssh.com) and implicit MAC/integrity algorithms, P-384 ECDSA host keys, and fixed rekey limits of 1 hour and 0.5 Gigabytes (whichever comes first).

Section 3.15 “RADIUS Configuration” of the Admin Guide states For both its RADIUS and syslog configuration, the TOE offers a fixed configuration supporting, IKEv2, tunnel-mode, NAT-T support, and a fixed set of IKEv2 and ESP algorithms: AES-256-CBC or GCM, SHA-384, and DH Groups 19 and 20 (ECP-256 and 384). The TOE allows no administrator configuration of this fixed configuration other than the SA lifetimes. The TOE allows and administrator to configure the RADIU/Syslog SA lifetime between 3-24 hours and the Child SA Lifetime between 1-8 hours.

Component Testing Assurance Activities: 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.

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.2.

2.2.13 Cryptographic Operation (Signature Generation and Verification) (NDcPP22e:FCS_COP.1/SigGen)

2.2.13.1 NDcPP22e:FCS_COP.1.1/SigGen

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

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

Section 6.2 of the ST states the TOE generates ECDSA signatures (during IPsec/IKE peer authentication) using curves P-384. The TOE also verifies ECDSA signatures during IKE peer authentication and when verifying trusted updates.

Component Guidance Assurance Activities: 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.

The evaluator confirmed that the TOE allows only ECDSA P-384 and allows no configuration.

Section 3.9 “Provisioning the TOE” of the Admin Guide states With this action the TOE will generate ECDSA P-384 private/public key pairs, generate SAE-PK keys/password, and upload and save a X.509 certificate signing request (CSR) on the provisioning PC.

Section 3.12 “Admin Configuration” of the Admin Guide states Beyond SSH key exchange, the TOE utilizes a fixed configuration of AES-256-GCM (aes256-gcm@openssh.com) and implicit MAC/integrity algorithms, P-384 ECDSA host keys, and fixed rekey limits of 1 hour and 0.5 Gigabytes (whichever comes first).

Component Testing Assurance Activities: ECDSA Algorithm Tests

ECDSA FIPS 186-4 Signature Generation Test

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

ECDSA FIPS 186-4 Signature Verification Test

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

RSA Signature Algorithm Tests

Signature Generation Test

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.

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

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.

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

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.2.

2.2.14 IPsec Protocol - per TD0633 (NDcPP22e:FCS_IPSEC_EXT.1)

2.2.14.1 NDcPP22e:FCS_IPSEC_EXT.1.1

TSS Assurance Activities: 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.

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.

Section 6.2 of the ST explains the TOE implements the IPsec protocol as specified in RFC 4301, and the TOE matching incoming and outgoing traffic against packet filtering and SPD rules to determine whether traffic should bypass ESP encryption (BYPASS), have ESP encryption applied (PROTECT), or whether the traffic should be dropped (DISCARD). The administrator can configure these rules to a limited extent (as the TOE does not serve as a general-purpose VPN gateway).

The TOE routes all packets through the kernel's IPsec interface (ipsec0) when any of its VPNs are active. The kernel compares packets routed through this interface to the SPDs configured for the VPN to determine whether to PROTECT, BYPASS, or DISCARD each packet. The vendor designed the TOE, to allow no manual SPD configuration beyond specifying the IP addresses of the syslog and RADIUS server. Consequently, the TOE protects all traffic between VPN clients and TOE traffic to the syslog and RADIUS servers.

Guidance Assurance Activities: 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.

Section 3.14 “IP Filtering Configuration and SPD rules.” states, The TOE also allows an administrator to specify IPsec SPDs that govern the TOE’s VPNs. Because of the TOE’s dedicated use-case, the TOE can provide up to three different VPNs. First, the TOE provides a VPN that secures wireless client traffic (a wireless client is also referred to as an End User Device or EUD). The TOE creates an SPD that always encrypts all traffic to and from EUDs. Second, the administrator can configure the TOE to secure (with a VPN) export of its audit records to a syslog server. In this case, the TOE creates an SPD that protects TCP port 514 traffic sent to the configured syslog server. Finally, the administrator can configure a WPA-Enterprise mode and specify the corresponding RADIUS server. In this case, the TOE creates an SPD that protects the traffic sent to and from the RADIUS server. The TOE does not allow administrators to specify other SPD rules beyond these three.

Testing Assurance Activities: 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.

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.

The evaluator observed the TOE utilize a PROTECT, DISCARD, and BYPASS rules during the course of testing. Specifically, the evaluator observed a positive and negative case for PROTECT during FTP_ITC.1 syslog testing. And the evaluator observed positive and negative test cases for DISCARD and BYPASS during FPF_RUL_EXT.1 testing.

This was repeated for both the EUD and RADIUS/syslog VPNs.

2.2.14.2 NDcPP22e:FCS_IPSEC_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: The assurance activity for this element is performed in conjunction with the activities for FCS_IPSEC_EXT.1.1.

The evaluator uses the guidance documentation to configure the TOE to carry out the following tests:

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 create'' final entry that discards packets that do not match any previous entries). The evaluator sends the packet and observes that the packet was dropped.

See Test Case FCS_IPSEC_EXT.1.1 above.

2.2.14.3 NDcPP22e:FCS_IPSEC_EXT.1.3

TSS Assurance Activities: 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).

Section 6.2 of the ST states the TOE supports only tunnel mode.

Guidance Assurance Activities: The evaluator shall confirm that the guidance documentation contains instructions on how to configure the connection in each mode selected.

Section 3.8 of the Admin Guide states the TOE offers no other configuration options beyond the Algorithms and Lifetimes configuration (i.e., the TOE has a fixed configuration for all other IPsec aspects, including hash/HMAC algorithms [HMAC-SHA-384], IKEv2, tunnel mode, and NAT-T support).

Testing Assurance Activities: The evaluator shall perform the following test(s) based on the selections chosen:

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.

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.

Test 1: The TOE supports tunnel mode. The evaluator configured a VPN peer to require only tunnel mode. The evaluator then attempted to connect the IPsec VPN between the test peer and the TOE and observed via logs and a packet capture that the connection was successful.

This was repeated for both the EUD and RADIUS/syslog VPNs.

Test 2: Not applicable. The TOE does not support transport mode.

2.2.14.4 NDcPP22e:FCS_IPSEC_EXT.1.4

TSS Assurance Activities: The evaluator shall examine the TSS to verify that the algorithms are implemented. In addition, the evaluator ensures that the SHA-based HMAC algorithm conforms to the algorithms specified in FCS_COP.1(4)/KeyedHash Cryptographic Operations (for keyed-hash message authentication) and if the SHA-based HMAC function truncated output is utilized it must also be described.

Section 6.2 of the ST states the TOE implements both the AES-CBC-256 and AES-GCM-256 ciphers and implements the truncated HMAC-SHA-384 (which is truncated to 192 bits) algorithm for ESP integrity (the TOE uses HMAC-SHA-384 only when paired with the AES-CBC 256 cipher). The TOE's OpenSSL library implements the AES-CBC and HMAC-SHA-384 algorithms, while the TOE's Kernel Cryptography implements the AES-GCM algorithm.

Guidance Assurance Activities: The evaluator checks the guidance documentation to ensure it provides instructions on how to configure the TOE to use the algorithms selected.

EUD VPN: Section 3.8 of the Admin Guide states Only AES-GCM-256/IKEv2 DH Group 20 is available in CSfC Compliant mode. The additional modes AES-GCM-256/IKEv2 DH Group 19, AES-CBC-256/IKEv2 DH Group 20 and AES-CBC-256/IKEv2 DH Group 19 are available when the administrator leaves CSfC Compliant mode unchecked.

RADIUS/syslog VPNs: Section 3.15 of the Admin Guide states for both its RADIUS and syslog configuration, the TOE offers a fixed configuration supporting, IKEv2, tunnel-mode, NAT-T support, and a fixed set of IKEv2 and ESP algorithms: AES-256-CBC or GCM, SHA-384, and DH Groups 19 and 20 (ECP-256 and 384). The TOE allows no administrator configuration of this fixed configuration other than the SA lifetimes.

Testing Assurance Activities: 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.

Test - The evaluator configured the TOE for AES-CBC-256 and AES-GCM-256 and verified via logs and packet captures that the connection was successfully established for each algorithm. This was repeated for both the EUD and RADIUS/syslog VPNs.

2.2.14.5 NDcPP22e:FCS_IPSEC_EXT.1.5

TSS Assurance Activities: The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented.

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.

Section 6.2 of the ST states the TOE implements only IKEv2.

Guidance Assurance Activities: 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 for the following test (if selected).

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.

EUD VPN: Section 3.8 of the Admin Guide states The TOE offers no other configuration options beyond the Algorithms and Lifetimes configuration (i.e., the TOE has a fixed configuration for all other IPsec aspects, including hash/HMAC algorithms [HMAC-SHA-384], IKEv2, tunnel mode, and NAT-T support).

RADIUS/syslog VPNs: Section 3.15 of the Admin Guide states For both its RADIUS and syslog configuration, the TOE offers a fixed configuration supporting, IKEv2, tunnel-mode, NAT-T support, and a fixed set of IKEv2 and ESP algorithms: AES-256-CBC or GCM, SHA-384, and DH Groups 19 and 20 (ECP-256 and 384). The TOE allows no administrator configuration of this fixed configuration other than the SA lifetimes.

Testing Assurance Activities: 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.

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 1: The TOE does not claim IKEv1.

Test 2: The evaluator configured the connection between the TOE and test peer so that NAT was required - both devices were on separate class C networks with a NAT router bridging the two networks. The evaluator then attempted to connect the IPsec VPN between the test peer and the TOE expecting the connection to be successful regardless of the NAT routing. The evaluator initiated an IPsec connection and observed that the TOE correctly negotiated the NAT connection to establish a protected IPsec connection. This was repeated for both the EUD and RADIUS/syslog VPNs.

2.2.14.6 NDcPP22e:FCS_IPSEC_EXT.1.6

TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE uses AES-CBC-256 or AES-GCM-256 to encrypt its IKEv2 SAs. This matches the selections in the requirement.

Guidance Assurance Activities: The evaluator ensures that the guidance documentation describes the configuration of all selected algorithms in the requirement.

EUD VPN: Section 3.8 of the Admin Guide states Only AES-GCM-256/IKEv2 DH Group 20 is available in CSfC Compliant mode. The additional modes AES-GCM-256/IKEv2 DH Group 19, AES-CBC-256/IKEv2 DH Group 20 and AES-CBC-256/IKEv2 DH Group 19 are available when the administrator leaves CSfC Compliant mode unchecked.

RADIUS/syslog VPN: Section 3.15 of the Admin Guide states for both its RADIUS and syslog configuration, the TOE offers a fixed configuration supporting, IKEv2, tunnel-mode, NAT-T support, and a fixed set of IKEv2 and ESP algorithms: AES-256-CBC or GCM, SHA-384, and DH Groups 19 and 20 (ECP-256 and 384). The TOE allows no administrator configuration of this fixed configuration other than the SA lifetimes.

Testing Assurance Activities: 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.

Test - The evaluator configured IKEv2 (for AES-GCM-256 and AES-CBC-256) on the TOE. The evaluator then confirmed that the TOE could establish a session with each algorithm and that the tunnel successfully established with the selected algorithm. This was repeated for both the EUD and RADIUS/syslog VPNs.

2.2.14.7 NDcPP22e:FCS_IPSEC_EXT.1.7

TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE allows an administrator to provision the TOEs IKEv2 SA lifetime to a value between 3 and 24 hours. This matches the selections in the requirement.

Guidance Assurance Activities: 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.

EUD VPN: Section 3.8 of the Admin Guide states the following:

All detailed configuration values for the radio network IP addresses, Wi-Fi parameters, or VPN cert CN names will need to be set. Select the “Edit Config” button on the middle-right area of the menu.

Radio (Wired) Network Section:

IP Address – The IP address of the TOE’s Ethernet interface.

Netmask Bits – The number of bits in the netmask for the TOE’s Ethernet interface. This needs to match the subnet mask configured in the wired Ethernet interface.

Gateway – The default gateway address for the TOE’s Ethernet interface. The radio network IP addressing scheme needs to match the IP addressing scheme configured into the wired radio.

Wi-Fi Section:

SSID is hidden – check this box if you don’t want the SSID advertised.

SSID – The name advertised by the TOE for Wi-Fi connections.

IP Address – Address of the TOE’s Wi-Fi interface and it must end in “.1”. EUDs will be given DHCP address of “.2”, “.3”, etc.

Num Devices – The max number of EUDs to be connected to this RAP.

80211 Mode – The mode and channel bandwidth of the Wi-Fi configuration.

Channel – The Wi-Fi channel to be used.

Encryption – The Wi-Fi encryption to be used. Note: Using any of the CCMP-256 settings may result in a reduction of about 50% of the Wi-Fi throughput. Some EUDs do not support CCMP-256 mode only.

Transmit Power – The available transmit power settings (in 1dBm steps from min to max) based on the 80211 Mode selected above.

WPA3-SAE-PK only mode – check this box if the RAP should only accept Wi-Fi connections in WPA3-SAE-PK mode. This mode is enabled by default and is required for a CSfC compliant configuration. It can only be changed if the CSfC Compliant Mode checkbox under Menu is unchecked.

VPN Section:

X509 Cert CN – The name of the RAP’s x509 certificate CN.

Algorithms – List of available encryption algorithms. Only AES-GCM-256/IKEv2 DH Group 20 is available in CSfC Compliant mode. The additional modes AES-GCM-256/IKEv2 DH Group 19, AES-CBC-256/IKEv2 DH Group 20 and AES-CBC-256/IKEv2 DH Group 19 are available when the administrator leaves CSfC Compliant mode unchecked.

SA Lifetime – The security association (SA) lifetime for IKE Phase 1. Valid values are 3-24 hours via the pulldown selection.

Child SA Lifetime – This is the SA lifetime for the IPSec data path. Valid values are 1-8 hours via the pulldown selection.

The TOE offers no other configuration options beyond the Algorithms and Lifetimes configuration (i.e., the TOE has a fixed configuration for all other IPsec aspects, including hash/HMAC algorithms [HMAC-SHA-384], IKEv2, tunnel mode, and NAT-T support).

RADIUS/syslog VPN: Section 3.15 of the Admin Guide states The TOE allows and administrator to configure the RADIU/Syslog SA lifetime between 3-24 hours and the Child SA Lifetime between 1-8 hours.

Testing Assurance Activities: 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.'

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 a 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.

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.

Test 1: This test is not applicable as 'number of bytes' is not selected for IKE/Phase 1 exchanges.

Test 2: The evaluator configured the TOE to have a 24 hour IKE and 8 hour ESP limits and the VPN test peer was configured to have 25 hour IKE and 9 hour ESP limits. The evaluator then connected the IPsec VPN between the test peer and the TOE. The evaluator observed through logs and packet captures that the connection was successful and that the TOE rekeyed each time the configured time limit was reached. This was repeated for both the EUD and RADIUS/syslog VPNs.

2.2.14.8 NDcPP22e:FCS_IPSEC_EXT.1.8

TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE allows an administrator to provision the TOEs ESP SA lifetime to a value between 1 and 8 hours. This matches the selections in the requirement.

Guidance Assurance Activities: 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 8 hours. 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.

EUD VPN: Section 3.8 of the Admin Guide states the following:

All detailed configuration values for the radio network IP addresses, Wi-Fi parameters, or VPN cert CN names will need to be set. Select the “Edit Config” button on the middle-right area of the menu.

Radio (Wired) Network Section:

IP Address – The IP address of the TOE’s Ethernet interface.

Netmask Bits – The number of bits in the netmask for the TOE’s Ethernet interface. This needs to match the subnet mask configured in the wired Ethernet interface.

Gateway – The default gateway address for the TOE's Ethernet interface. The radio network IP addressing scheme needs to match the IP addressing scheme configured into the wired radio.

Wi-Fi Section:

SSID is hidden – check this box if you don't want the SSID advertised.

SSID – The name advertised by the TOE for Wi-Fi connections.

IP Address – Address of the TOE's Wi-Fi interface and it must end in ".1". EUDs will be given DHCP address of ".2", ".3", etc.

Num Devices – The max number of EUDs to be connected to this RAP.

80211 Mode – The mode and channel bandwidth of the Wi-Fi configuration.

Channel – The Wi-Fi channel to be used.

Encryption – The Wi-Fi encryption to be used. Note: Using any of the CCMP-256 settings may result in a reduction of about 50% of the Wi-Fi throughput. Some EUDs do not support CCMP-256 mode only.

Transmit Power – The available transmit power settings (in 1dBm steps from min to max) based on the 80211 Mode selected above.

WPA3-SAE-PK only mode – check this box if the RAP should only accept Wi-Fi connections in WPA3-SAE-PK mode. This mode is enabled by default and is required for a CSfC compliant configuration. It can only be changed if the CSfC Compliant Mode checkbox under Menu is unchecked.

VPN Section:

X509 Cert CN – The name of the RAP's x509 certificate CN.

Algorithms – List of available encryption algorithms. Only AES-GCM-256/IKEv2 DH Group 20 is available in CSfC Compliant mode. The additional modes AES-GCM-256/IKEv2 DH Group 19, AES-CBC-256/IKEv2 DH Group 20 and AES-CBC-256/IKEv2 DH Group 19 are available when the administrator leaves CSfC Compliant mode unchecked.

SA Lifetime – The security association (SA) lifetime for IKE Phase 1. Valid values are 3-24 hours via the pulldown selection.

Child SA Lifetime – This is the SA lifetime for the IPSec data path. Valid values are 1-8 hours via the pulldown selection.

The TOE offers no other configuration options beyond the Algorithms and Lifetimes configuration (i.e., the TOE has a fixed configuration for all other IPsec aspects, including hash/HMAC algorithms [HMAC-SHA-384], IKEv2, tunnel mode, and NAT-T support).

RADIUS/syslog VPN: Section 3.15 of the Admin Guide states The TOE allows and administrator to configure the RADIU/Syslog SA lifetime between 3-24 hours and the Child SA Lifetime between 1-8 hours.

Testing Assurance Activities: 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.'

Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

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 a 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.

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.

Test 1 - The TOE does not claim support for "number of bytes".

Test 2 - This test was performed as part of FCS_IPSEC_EXT.1.7 test 2.

2.2.14.9 NDcPP22e:FCS_IPSEC_EXT.1.9

TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE supports key exchange groups DH19 and DH20 and generates a secret “x” of size 256 and 384 bits, respectively. The length meets the SFR.

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.14.10 NDcPP22e:FCS_IPSEC_EXT.1.10

TSS Assurance Activities: 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.

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.

Section 6.2 of the ST states the TOE generates IKEv2 nonces using its DRBG, and for DH19 and DH20, the TOE generates a 256 bit and 384 bit long nonce, respectively. Those lengths represent double the security strength..

Guidance Assurance Activities: None Defined

Testing Assurance Activities: 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.

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.

See TSS assurance activity above.

2.2.14.11 NDcPP22e:FCS_IPSEC_EXT.1.11

TSS Assurance Activities: 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.

Section 6.2 of the ST states that by default, the TOE supports DH20 (ECP-384) but allows an administrator to provision the TOE to support DH19 (ECP-256) as well.

Guidance Assurance Activities: The evaluator ensures that the guidance documentation describes the configuration of all algorithms selected in the requirement.

EUD VPN: Section 3.8 of the Admin Guide states Only AES-GCM-256/IKEv2 DH Group 20 is available in CSfC Compliant mode. The additional modes AES-GCM-256/IKEv2 DH Group 19, AES-CBC-256/IKEv2 DH Group 20 and AES-CBC-256/IKEv2 DH Group 19 are available when the administrator leaves CSfC Compliant mode unchecked.

RADIUS/syslog VPN: Section 3.15 of the Admin Guide states For both its RADIUS and syslog configuration, the TOE offers a fixed configuration supporting, IKEv2, tunnel-mode, NAT-T support, and a fixed set of IKEv2 and ESP algorithms: AES-256-CBC or GCM, SHA-384, and DH Groups 19 and 20 (ECP-256 and 384). The TOE allows no administrator configuration of this fixed configuration other than the SA lifetimes.

Testing Assurance Activities: 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.

Test 1 - The evaluator made an IPsec connection to an IPsec peer using each of the claimed DH groups. The evaluator was able to capture each DH group using a packet capture. This was repeated for both the EUD and RADIUS/syslog VPNs.

2.2.14.12 NDcPP22e:FCS_IPSEC_EXT.1.12

TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE only allows SA cipher strengths of 256 bits, hence the TOE's design inherently prevents a situation in where the ESP SA cipher strength exceeds that of theIKEv2 SA.

Guidance Assurance Activities: None Defined

Testing Assurance Activities: 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.

b) Test 2: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish a 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.

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.

d) Test 4: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish a 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.

Test 1 - The evaluator made an IPsec connection to an IPsec peer using each of the claimed hash functions identified in the requirements. The evaluator verified via packet capture and logs that the connection was successful with each of the claimed functions.

Test 2 - The evaluator configured a test peer to use a 128 bit key size for IKE and a 256 bit key size for ESP. The evaluator then attempted to connect the IPsec VPN between the test peer and the TOE and confirmed that the connection was rejected by the TOE.

Test 3 - The evaluator attempted to establish a connection with an unsupported algorithm/hash combination. The connection attempt failed.

Test 4 - This test was performed as part of test 3.

This was repeated for both the EUD and RADIUS/syslog VPNs.

2.2.14.13 NDcPP22e:FCS_IPSEC_EXT.1.13

TSS Assurance Activities: 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(2)/SigGen Cryptographic Operations (for cryptographic signature).

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.

Section 6.2 of the ST states the TOE supports ECDSA certificates for IKE peer authentication.

This is consistent with the algorithms as specified in NDcPP:FCS_COP.1/SigGen.

Guidance Assurance Activities: 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.

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.

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'.

Section 3.9 of the Admin Guide discusses setting up the TOE to use an ECDSA certificate. It discusses creating the CSR and section 3.10 discusses importing the certificate (which automatically sets it as trusted).

Testing Assurance Activities: For efficiency sake, the testing is combined with the testing for FIA_X509_EXT.1, FIA_X509_EXT.2 (for IPsec connections), and FCS_IPSEC_EXT.1.1.

Testing in NDcPP22e_FCS_IPSEC_EXT.1.3-t1 demonstrates that a successful IPsec connection using an ECDSA certificate can be established.

2.2.14.14 NDcPP22e:FCS_IPSEC_EXT.1.14

TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE supports Distinguished Name checking of VPN client certificates and checking of SAN:IPv4 (IPv4 addresses in the Subject Alternative Name) for syslog and RADIUS peer certificates.

Section 6.2 of the ST states the TOE supports Distinguished Name checking of VPN client certificates.

Guidance Assurance Activities: 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.

EUD VPN: Section 3.11 of the Admin Guide states For the TOE's EUD VPN network, an administrator must configure the TOE with EUD DNs. The administrator must do this after the TOE has been provisioned (as in the previous steps), and should use the DPA GUI to enter the list of EUD Distinguished Names (DN) or Common Names (CN) for all EUDs that will connect to this TOE.

RADIUS/syslog VPN: Section 3.15 of the Admin Guide states that For both its RADIUS and syslog configuration, the TOE uses the administrator defined server IPv4 address as the reference identifier. The TOE will inspect the peer's IKE authentication certificate to ensure that it contains an SAN:IPv4 field containing the administrator configured server IP address and reject the certificate otherwise.

Testing Assurance Activities: 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.

The evaluator shall perform the following tests:

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.

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.

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 ''. 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.

b) Configure the peer's reference identifier on the TOE (per the administrative guidance) to match the CN without the '' and verify that IKE authentication fails.

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.

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.

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.

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.

b) Append '' to a non-CN field of an otherwise authorized DN.

Test 1: Not applicable: the ST selected no CN/identifier type combinations.

Test 2:

(Part 1) The evaluator configured strongswan on a test peer to use an authentication certificate with the correct SAN: IP address, DNS address (FQDN), and user FQDN. The evaluator then attempted to connect the IPsec VPN between the test peer and observed the TOE accept each connection.

(Part 2), the test does not apply as the TOE does not support the Common Name (CN) for reference identifier, only Subject Alternative Name (SAN).

Test 3: Not applicable: the ST selected no CN/identifier type combinations.

Test 4:

(Part 1) For this test, the evaluator alternately configured the TOE to look for each of the supported SAN reference identifiers. The evaluator then configured the strongswan VPN peer to use a certificate that would present an incorrect SAN reference identfier and a correct CN reference identifier. In each case, the evaluator then attempted to connect the IPsec VPN between the test peer and the TOE expecting the connection to be rejected.

(Part 2) – the test does not apply as the TOE does not support the Common Name (CN) for reference identifier, only Subject Alternative Name (SAN).

Test 5: The evaluator configured a test peer to send a certificate with a valid DN. The evaluator then initiated a connection between the TOE and the test peer and confirmed that the connection was successful.

Test 6a: The evaluator configured a test server to first send a certificate with an authorized DN, and then a nearly identical certificate but with a DN containing a duplicate CN. In each case, the evaluator attempted to establish an IPsec connection and confirmed that the TOE rejected the certificate with the duplicate CN.

Test 6b: The evaluator configured a test server to first sent a certificate with an authorized DN, and then a nearly identical certificate in which the evaluator appended '\0' to a non-CN field of an otherwise authorized DN. In each case, the evaluator attempted to establish an IPsec connection and confirmed that the TOE rejected the certificate with the DN containing a null character.

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.2.15 IPsec Protocol - per TD0657 (VPNGW12:FCS_IPSEC_EXT.1)

Please note that for the VPNGW12 only defines a modified version of 1.14, and for all other SFRs, the VPNGW refers to NDcPP22e. The AA verdicts for all the VPNGW12 FCS_IPSEC_EXT.1.x have been combined into the verdicts to NDcPP22e:FCS_IPSEC_EXT.1.x above.

2.2.15.1 VPNGW12:FCS_IPSEC_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.2 VPNGW12:FCS_IPSEC_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.3 VPNGW12:FCS_IPSEC_EXT.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.4 VPNGW12:FCS_IPSEC_EXT.1.4

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.5 VPNGW12:FCS_IPSEC_EXT.1.5

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.6 VPNGW12:FCS_IPSEC_EXT.1.6

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.7 VPNGW12:FCS_IPSEC_EXT.1.7

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.8 VPNGW12:FCS_IPSEC_EXT.1.8

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.9 VPNGW12:FCS_IPSEC_EXT.1.9

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.10 VPNGW12:FCS_IPSEC_EXT.1.10

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.11 VPNGW12:FCS_IPSEC_EXT.1.11

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.12 VPNGW12:FCS_IPSEC_EXT.1.12

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.13 VPNGW12:FCS_IPSEC_EXT.1.13

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.15.14 VPNGW12:FCS_IPSEC_EXT.1.14

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: All existing activities regarding 'Pre-shared keys' apply to all selections including pre-shared keys. If any selection with 'Pre-shared keys' is included, the evaluator shall check to ensure that the TSS describes how the selection works in conjunction with the authentication of IPsec connections.

Not applicable. The TOE does not select Pre-shared keys.

Component Guidance Assurance Activities: If any selection with 'Pre-shared Keys' is selected, the evaluator shall check that the operational guidance describes any configuration necessary to enable any selected authentication mechanisms.

Not applicable. The TOE does not select Pre-shared keys.

Component Testing Assurance Activities: None Defined

2.2.16 Random Bit Generation (NDcPP22e:FCS_RBG_EXT.1)

2.2.16.1 NDcPP22e:FCS_RBG_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.16.2 NDcPP22e:FCS_RBG_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: Documentation shall be produced - and the evaluator shall perform the activities - in accordance with Appendix D of [NDcPP].

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.

Section 6.2 of the ST states the TOE seeds its AES-256 CTR_DRBG using a 384-bit seed from a hardware entropy source.

Component Guidance Assurance Activities: Documentation shall be produced - and the evaluator shall perform the activities - in accordance with Appendix D of [NDcPP].

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

Section 3.9 of the Admin Guide states Finally, note that the TOE includes an internal DRBG that it uses when generating key pairs (whether for the CSR ultimately used for IKE authentication or for SSH, IKE, and WPA key exchange). The administrator need not and cannot configure the TOE’s DRBG functionality; the TOE automatically seeds and uses its internal DRBG appropriately.

Component Testing Assurance Activities: The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration.

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).

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

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

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

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

Personalization string: The length of the personalization string must be <= 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.

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.2.

2.2.17 SSH Server Protocol - per TD0631 (NDcPP22e:FCS_SSHS_EXT.1)

2.2.17.1 NDcPP22e:FCS_SSHS_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.17.2 NDcPP22e:FCS_SSHS_EXT.1.2

TSS Assurance Activities: 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).

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.

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. (TD0631 applied)

Section 6.2 of the TSS states TOE supports only ECDSA client certificates and matches an SSH client's presented public key with one within the TOE's authorized_keys file. This description is consistent with NDcPP22e:FCS_COP.1/SigGen. The TOE also supports password-based authentication, and in the absence of a presented public key, the TOE prompts the client to supply a username and password.

Guidance Assurance Activities: None Defined

Testing Assurance Activities: Test objective: The purpose of these tests is to verify server supports each claimed client authentication method.

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.

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.

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.

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.

(TD0631 applied)

Test 1: This evaluator attempted to connect to the TOE using a SSH client using claimed pubkey(s) configured on the TOE. The evaluator found that the connection succeeded and the TOE accepted the pubkey authentication without falling back to password authentication.

Test 2: The evaluator attempted to connect to the TOE using a SSH client using a pubkey not configured on the TOE. The evaluator found that an unconfigured pubkey would not be used to establish an SSH session and the TOE would revert back to password authentication.

Test 3: The evaluator attempted to connect to the TOE using a SSH client alternately using the correct and incorrect password. The evaluator found that only the correct password would yield a successful SSH session.

Test 4: This test was performed as part of Test 3 as described above.

2.2.17.3 NDcPP22e:FCS_SSHS_EXT.1.3

TSS Assurance Activities: The evaluator shall check that the TSS describes how 'large packets' in terms of RFC 4253 are detected and handled.

Section 6.2 of the ST states the TOE inspects incoming SSH packets to check for those larger than 263,144 bytes in size and drops such packets.

Guidance Assurance Activities: None Defined

Testing Assurance Activities: The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

The evaluator created and sent a packet to the TOE that was larger than the maximum packet size of 263,144 bytes. The evaluator observed that the TOE rejected the packet and the connection was closed.

2.2.17.4 NDcPP22e:FCS_SSHS_EXT.1.4

TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE supports the AES-256-GCM cipher mode. This matches the selection in the requirement.

Guidance Assurance Activities: 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).

The evaluator found that the TOE offers and requires no configuration of SSH ciphers to conform to the requirements.

Section 3.12 of the Admin Guide states Beyond SSH key exchange, the TOE utilizes a fixed configuration of AES-256-GCM (aes256-gcm@openssh.com) and implicit MAC/integrity algorithms, P-384 ECDSA host keys, and fixed rekey limits of 1 hour and 0.5 Gigabytes (whichever comes first).

Testing Assurance Activities: 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.

The evaluator attempted to establish an SSH connection with each of the following SSH algorithm to encrypt the session: AES256-GCM. The evaluator captured packets associated with the connection attempt and observed that using this algorithm the evaluator was able to successfully connect to the TOE.

2.2.17.5 NDcPP22e:FCS_SSHS_EXT.1.5

TSS Assurance Activities: 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. (TD0631 applied)

Section 6.2 of the ST states the TOE supports an ECDSA host key (size P-384) for when the TOE uses pubkey authentication to authenticate to connecting SSH clients. This matches the selection in the requirement.

Guidance Assurance Activities: 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).

Section 2.5 of the Admin Guide explains how to generate an ECDSA key with P-384 to meet the requirement.

Testing Assurance Activities: Test objective: This test case is meant to validate that the TOE server will support host public keys of the claimed algorithm types.

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.

Has effectively been moved to FCS_SSHS_EXT.1.2.

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.

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.

(TD0631 applied)

Test 1: The evaluator attempted to connect to the TOE using a SSH client alternately using each of the authentication algorithms that can be claimed to determine which ciphers are supported with successful connections. The evaluator confirmed that the TOE supports the ecdsa-sha2-nistp384 algorithm as claimed. The evaluator followed up with a disallowed authentication algorithm and confirmed that it was not accepted.

Test 2: This test was performed as part of Test 1.

2.2.17.6 NDcPP22e:FCS_SSHS_EXT.1.6

TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE supports the "implicit" mode of integrity associated with AES-GCM. This matches the selection in the requirement.

Guidance Assurance Activities: 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).

Section 3.12 of the Admin Guide states The TOE always supports ecdh-sha2-nistp384 for SSH key exchange, and the administrator can additionally enable support for SSH key exchange using ecdh-sha2-nistp256 by unchecking CSfC mode, and then, from the Admin tab, checking the "Add SSH Key Exchange Algorithm ecdh-sha2-nistp256" checkbox, closing, and applying the configuration.

Beyond SSH key exchange, the TOE utilizes a fixed configuration of AES-256-GCM (aes256-gcm@openssh.com) and implicit MAC/integrity algorithms, P-384 ECDSA host keys, and fixed rekey limits of 1 hour and 0.5 Gigabytes (whichever comes first).

Testing Assurance Activities: 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.

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.

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 the ST selection. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.

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.

These tests are not applicable, as the FCS_SSHS_EXT.1.6 selections in the ST do not include an HMAC or AED_AES_*_GCM algorithm.

2.2.17.7 NDcPP22e:FCS_SSHS_EXT.1.7

TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE supports ecdh-sha2-nistp256 and ecdh-sha2-nistp384. This is consistent with the requirement selections.

Guidance Assurance Activities: 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.

Section 3.12 of the Admin Guide states The TOE always supports ecdh-sha2-nistp384 for SSH key exchange, and the administrator can additionally enable support for SSH key exchange using ecdh-sha2-nistp256 by unchecking CSfC mode, and then, from the Admin tab, checking the "Add SSH Key Exchange Algorithm ecdh-sha2-nistp256" checkbox, closing, and applying the configuration.

Beyond SSH key exchange, the TOE utilizes a fixed configuration of AES-256-GCM (aes256-gcm@openssh.com) and implicit MAC/integrity algorithms, P-384 ECDSA host keys, and fixed rekey limits of 1 hour and 0.5 Gigabytes (whichever comes first).

Testing Assurance Activities: 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.

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.

Test 1 - This test was performed as part of Test 2. The evaluator attempted to connect to the TOE using DiffieHellman-Group1. The TOE rejected the attempt as expected.

Test 2 - The evaluator attempted to connect to the TOE using a SSH client alternately using an unallowable key exchange algorithm which fails and then each of the key exchange algorithms that can be claimed to determine which ciphers are supported with successful connections. The evaluator attempted to connect to the TOE using an SSH client with each claimed key exchange method: DH group 19 and DH20 and confirmed that the connections were successful.

2.2.17.8 NDcPP22e:FCS_SSHS_EXT.1.8

TSS Assurance Activities: 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.

Section 6.2 of the ST states the TOE supports rekey limits of 1 hour and 0.5 Gigabyte and initiates a rekey when it encounters either threshold.

Guidance Assurance Activities: 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.

Section 3.12 of the Admin Guide states Beyond SSH key exchange, the TOE utilizes a fixed configuration of AES-256-GCM (aes256-gcm@openssh.com) and implicit MAC/integrity algorithms, P-384 ECDSA host keys, and fixed rekey limits of 1 hour and 0.5 Gigabytes (whichever comes first).

Testing Assurance Activities: 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.

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).

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.

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).

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).

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.

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(3)/Functions).

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.

Test - The evaluator attempted to connect to the TOE using an SSH client generating 1GB of data and confirmed via logs and packet captures that the rekey happened before the threshold was reached. The evaluator then attempted to connect to the TOE using a SSH client waiting an hour and confirmed via logs and packet captures that the rekey happened before the 1 hour threshold was reached.

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.3 Identification and authentication (FIA)

2.3.1 802.1X Port Access Entity (Authenticator) Authentication (WLANAS10:FIA_8021X_EXT.1)

2.3.1.1 WLANAS10:FIA_8021X_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.3.1.2 WLANAS10:FIA_8021X_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.3.1.3 WLANAS10:FIA_8021X_EXT.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: In order to show that the TSF implements the 802.1X-2010 standard correctly, the evaluator will 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

- For each identified section, any non-conformance is identified and described, including a justification for the non-conformance

Because the connection to the RADIUS server will be contained in an IPsec or RadSec (TLS) tunnel, 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 will 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.

Section 6.3 of the ST states the TOE adheres to the 802.1X-2010 standard in supporting EAPOL for wireless authentication (IEEE 802.11) and does not support other 802.1X methods included in the standard (e.g., Token Ring, FDDI, MACsec, etc.). The vendor has performed research and testing during product development to ensure compatibility and interoperability.

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: The evaluator will perform the following tests:

Test 1: The evaluator will demonstrate that a wireless client has no access to the test network. After successfully authenticating with a RADIUS server through the TOE, the evaluator will demonstrate that the wireless client does have access to the test network.

Test 2: The evaluator will demonstrate that a wireless client has no access to the test network. The evaluator will 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.

Test 3: The evaluator will demonstrate that a wireless client has no access to the test network. The evaluator will 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.

Note: Tests 2 and 3 above are not tests that 'EAP-TLS works,' although that is 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 demonstrates the enforcement of FIA_8021X_EXT.1.3.

Test 1: The evaluator configured the TOE for wireless radius authentication. The evaluator then attempted to access the network prior to completing authentication and verified the attempt failed. The evaluator then attempted to access the network after completing authentication and verified the attempt succeeded.

Test 2: The evaluator attempted a connection with an invalid client certificate and verified that the connection was rejected and the client was unable to access the test network.

Test 3: The evaluator attempted a radius authentication connection with an invalid RADIUS certificate and verified that the connection was rejected and the client was unable to access the test network.

2.3.2 Authentication Failure Management (NDcPP22e:FIA_AFL.1)

2.3.2.1 NDcPP22e:FIA_AFL.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.3.2.2 NDcPP22e:FIA_AFL.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

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).

Section 6.3 of the ST states the TOE allows the administrator to configure an admin lock out time between 5 to 120 minutes. The TOE tracks failed password authentication attempts for each user and will lock out the user for the administrator configured time, or until another administrator unlocks the user.

The TOE ensures that while failed logins results in lockout of remote users (administrators), the TOE does not lock out local administrators (those accessing the TOE via its USB interface) after failed logins.

Component Guidance Assurance Activities: 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.

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.

Section 3.3 of the Admin Guide explains that the script unlock-admin-acct can be used to unlock a user account or the administrator can wait until the timer expires. To avoid total lockout on the TOE, the default admin account will not be locked out on missed attempts; however, the default admin account will only be able to log in on the USB interface to the TOE.

Section 3.12 of the Admin Guide states An administrator can use the DPA GUI (Edit Config->Admin [Tab]->Account Lockout Time (min)) to set the TOE’s Account Time Period in minutes to a value between 5 and 120 and to set the TOE’s Password Retry Times (Edit Config->Admin [Tab]->Password Retry Times) to a value between 3 and 10.

Component Testing Assurance Activities: 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.

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).

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.

Test 1: The evaluator configured the TOE for a 3 failed authentication attempt and 5 minute lockout duration. The evaluator then attempted to connect to a session and verified after 3 failures the session was locked. After waiting the 5 minute lockout period the evaluator was able to reconnect. This was repeated with a 5 failed authentication attempts and 7 minute lockout duration with the same results.

Test 2: This was performed as part of test 1, where the evaluator waited until just before the configured timeout setting and attempted a login with valid credentials, and confirmed that this attempt was unsuccessful. The evaluator then waited until just after the configured time period and attempted a login with valid credentials and confirmed that the attempt was successful. This test was completed for both the 5 minute time out and the 7 minute time out. Additionally the evaluator preformed testing that confirmed that the vendor's guidance allows a second administrative account to successfully unlock a different, locked admin account.

2.3.3 Password Management (NDcPP22e:FIA_PMG_EXT.1)

2.3.3.1 NDcPP22e:FIA_PMG_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.3 of the ST states the TOE supports passwords with a minimum and maximum length of between 6 and 16 characters consisting of upper and lower case letters, numbers and the following special characters: ['!', '@', '#', '$', '%', '^', '&', '*', '(', ')', [= + - _ ` ~ |] [{ } ˜ ' ; : / ? . > , <]]

Component Guidance Assurance Activities: 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.

Section 3.12 of the AGD goes over how to set the minimum password length and what values are acceptable as well as stating that Administrative passwords may contain any of the 95 ASCII printable characters, and administrators can specify string passwords as a series of any of the 95 ASCII printable characters.

Component Testing Assurance Activities: The evaluator shall perform the following tests.

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.

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.

The evaluator configured passwords to meet all of the rules in the test procedures and verified that passwords that do not meet the requirements were rejected.

2.3.4 Re-Authenticating (WLANAS10:FIA_UAU.6)

2.3.4.1 WLANAS10:FIA_UAU.6.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: The evaluator will attempt to change their password as directed by the operational guidance. While making this attempt, the evaluator will verify that re-authentication is required. If other re-authentication conditions are specified, the evaluator will cause those conditions to occur and verify that the TSF re-authenticates the authenticated user.

Test 1: The evaluator followed operational guidance to change the administrator password, then verified that the session immediately required re-authentication.

2.3.5 Protected Authentication Feedback (NDcPP22e:FIA_UAU.7)

2.3.5.1 NDcPP22e:FIA_UAU.7.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: 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.

The TOE obscures authentication data by default so no guidance is required.

Component Testing Assurance Activities: 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.

Test 1: See NDcPP22e:FIA_UIA_EXT.1.

2.3.6 Password-based Authentication Mechanism (NDcPP22e:FIA_UAU_EXT.2)

2.3.6.1 NDcPP22e:FIA_UAU_EXT.2.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

See NDcPP22e:FIA_UIA_EXT.1.

Component Guidance Assurance Activities: 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.

See NDcPP22e:FIA_UIA_EXT.1

Component Testing Assurance Activities: 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.

See NDcPP22e:FIA_UIA_EXT.1

2.3.7 User Identification and Authentication (NDcPP22e:FIA_UIA_EXT.1)

2.3.7.1 NDcPP22e:FIA_UIA_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.3.7.2 NDcPP22e:FIA_UIA_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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'.

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.

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.

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.

Section 6.3 of the ST states the TOE provides both username/password as well as pubkey authentication for administrators (who connect via SSH). The TOE allows no actions prior to successful authentication (and successful authentication consists of either a valid pubkey authentication or a correct username/password combination). The TOE provides the same SSH authentication to local and remote administrators; however, access through the TOE's Ethernet port constitutes remote administration and access through the TOE's USB interface constitutes local administration.

Component Guidance Assurance Activities: 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.

Section 2.5 of the Admin Guide explains how to setup SSH to permit connections. All administration is done via SSH so this a complete description. Section 3.2 of the Admin Guide explains how to setup the account and log into the TOE.

Component Testing Assurance Activities: 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.

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.

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.

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.

The TOE offers the following user interfaces where authentication is provided.

- SSH using passwords

Test 1 - Using each interface the evaluator performed an unsuccessful and successful logon of each type using bad and good credentials respectively.

Test 2 - Using each interface the evaluator was able to observe the TOE displayed a banner to the user before login.

Test 3 - Using each interface the evaluator found that, prior to login, no functions were available to the administrator with the exception of acknowledging the banner.

Test 4 - The TOE is not distributed, thus tests 1 through 3 above test the only TOE component.

2.3.8 X.509 Certificate Validation (NDcPP22e:FIA_X509_EXT.1/Rev)

2.3.8.1 NDcPP22e:FIA_X509_EXT.1.1/Rev

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: 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.

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

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.

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.

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.)

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.)

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.)

h) The following tests are run when a minimum certificate path length of three certificates is implemented.

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 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 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.

(TD0527 12/2020 update applied)

Test 1a & 1b -- The evaluator configured the TOE and a peer with valid certificates. The evaluator then attempted to make a connection between the peer devices. A successful connection was made. The evaluator then configured a server certificate with an invalid certification path by deleting an intermediate root CA so that the certificate chain was invalid because of a missing (or deleted) certificate. The connection between the peers was refused.

Test 2 -- The evaluator attempted to make a connection between the TOE and a test server. The test server then presented an expired certificate during the negotiation resulting in a failed connection.

Test 3 -- The evaluator attempted to make a connection between the TOE and a test server. The test server then presented a certificate during the TLS negotiation where the certificate was valid. A packet capture was obtained of this TLS negotiation which shows that the connection was successful. The evaluator revoked certificates in the chain (individually) and attempted the same connection. The attempt after revoking the certificate was not successful.

Test 4 -- The evaluator configured an CRL responder to present a certificate that does not have the CRL signing purpose. The evaluator established a connection between the TOE and a test server such that the TOE receives CRL response signed by the invalid certificate and ensured that the connection was not negotiated successfully.

Test 5 -- The evaluator configured a test server to present a certificate that had a byte in the first eight bytes modified to the TOE. The evaluator then attempted to make a connection between the peer devices. When the TOE attempted to connect, the connection failed.

Test 6 -- The evaluator configured a test server to present a certificate that had a byte in the last eight bytes modified to the TOE. The evaluator then attempted to make a connection between the peer devices. When the TOE attempted to connect, the connection failed.

Test 7 -- The evaluator configured a test server to present a certificate that had a byte in the public key of the certificate modified to the TOE. The evaluator then attempted to make a connection between the peer devices. When the TOE attempted to connect, the connection was refused.

Test 8a - The evaluator configured the test peer to send a valid chain of ECDSA certificates where only the root CA is located in the trust store, and where the ECDSA certificates have a supported named curve. The TOE accepted the certificate as valid.

Test 8b - The evaluator then configured the test peer to send a valid chain of ECDSA certificates where only the root CA is located in the trust store, and where an intermediate CA's ECDSA certificate has an explicit curve. The TOE rejected the certificate as invalid.

Test 8c - The evaluator attempted to load an intermediate CA ECDSA certificate with a supported named curve into the TOE trust store and observed that the certificate can be loaded. The evaluator attempted to load an intermediate CA ECDSA certificate with an explicit curve into the TOE trust store and observed that the certificate could not be loaded.

The evaluator repeated these tests for both the EUD and RADIUS/syslog VPNs.

2.3.8.2 NDcPP22e:FIA_X509_EXT.1.2/Rev

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: 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.

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). 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).

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).

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).

Test 1: The evaluator configured a test server to present a certificate chain containing a CA certificate lacking the basicConstraints extension. The evaluator then attempted a connection between the TOE and the test server and observed that the TOE rejected the connection.

Test 2: The evaluator configured a test server to present a certificate chain containing a CA certificate having the basicConstraints section but with the cA flag not set (i.e., FALSE). The evaluator then attempted a connection between the TOE and the test server and observed that the TOE rejected the connection.

The evaluator repeated these tests for both the EUD and RADIUS/syslog VPNs.

Component TSS Assurance Activities: 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).

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.

Section 6.3 of the ST explains the TOE checks the validity (expiration as well as checking for explicit curves) of X.509 certificates during loading and during IKE authentication. The TOE enforces no requirements on extendedKeyUsage fields of the peer's certificate. The TOE checks revocation by checking the Certificate Revocation List (CRL) of the issuing CA. The TOE attempts to check certificate revocation status via CRLs when performing IKE peer/client authentication. The TOE requires that the IKE peer/client always provide a full certificate chain.

Component Guidance Assurance Activities: 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.

Section 3.11 of the Admin Guide states Once the TOE receives an IKE peer's certificate, the TOE performs a series of check to ensure validity, including checking whether the certificate remains valid or has expired, checks the peer certificate chains to a trusted root CA (that the administrator has already imported into the TOE), the TOE checks for other, required certificate properties (including the presence of the basicconstraints section and a cA:true flag for CA certs, cRLsign key usage for CA certs; however the TOE does not require any extendedKeyUsage fields be preset), and finally checks all certificates in the peer's chain for revocation using CRLs (obtaining those CRLs from the specified CDP information contained within the certificates.

Component Testing Assurance Activities: None Defined

2.3.9 X.509 Certificate Authentication (NDcPP22e:FIA_X509_EXT.2)

2.3.9.1 NDcPP22e:FIA_X509_EXT.2.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.3.9.2 NDcPP22e:FIA_X509_EXT.2.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates.

The evaluator shall examine the TSS to confirm that it describes the 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.

Section 6.3 of the ST states the administrator can configure the root CA used during IKE authentication. If the TOE cannot establish a connection with a revocation server to check the status of a certificate in question, the TOE will not accept the certificate (the TOE rejects the certificate as invalid).

Component Guidance Assurance Activities: 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.

Section 2.4 of the Admin Guide states

- TOE configurations always have strict CRL checking turned on so valid CRLs need to be available via a Certificate Distribution Point (CDP) or imported when configuring each TOE.
  - CRL files must be in PEM format.
  - The TOE automatically rejects any certificate where the TOE cannot determine the renovation status (e.g., the TOE cannot reach the revocation server)
  - The administrator need not configure anything and cannot change or override this aspect of the TOE's behavior.

In relation to configuring a certificate for the TOE's VPN Client (EUD) network, section 3.9 of the Admin Guide states that after entering all the configuration data, close the configuration tab select "Provision" in the DPA GUI to push the configuration data to the TOE. With this action the TOE will generate ECDSA P-384 private/public key pairs, generate SAE-PK keys/password, and upload and save a X.509 certificate signing request (CSR) on the provisioning PC. During provisioning, a status bar will show progress and the status line at the bottom of the screen displays current activities being performed. The location of the CSR that needs to be signed by a Certificate Authority (CA) is displayed at the bottom of the UI in the status area.

The above CSR file that was just generated needs to be sent to a CA to be signed. The signed certificate will then be brought back to the Linux PC and imported into the TOE along with the CA certificate chain of the signing CAs. The explanation of CAs, X.509 certificate signing and public key infrastructure (PKI) tools and processes is beyond the scope of this document. Operational Guidance for the TOE.

Furthermore, section 3.10 of the Admin Guide states

Once a signed certificate file is available and the CA certificate chain file has been copied to the provisioning computer (be sure to note which folder they get saved), they can be imported into the TOE through the “Menu” pulldown on the top left of the main provisioning screen.

In relation to configuring a certificate for audit/syslog, section 3.13 of the Admin Guide states

In addition to these settings the X.509v3 certificates need to be created and imported into the TOE. This is done via the menu options available when on this tab to Generate New Certificate Request, Import Signed Certificate, Import Trusted CA Certs and Import CRL File.

In relation to configuring a certificate for RADIUS, Section 3.15 of the Admin Guide states

The bottom part of this tab is used to configure the X.509v3 certificates for the VPN that forms the trusted channel for the RADIUS data. This is done via the menu options available when on this tab to Generate New Certificate Request, Import Signed Certificate, Import Trusted CA Certs and Import CRL File.

Component Testing Assurance Activities: The evaluator shall perform the following test for each trusted channel:

The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the guidance documentation to determine that all supported administrator-configurable options behave in their documented manner.

The evaluator first performed a control test to verify that a successful connection was made using an authentication certificate with valid/accessible revocation servers sent from a test peer to the TOE. The evaluator then configured the TOE to ensure the revocation server was unreachable and verified that the connection succeeded.

2.3.10 X.509 Certificate Authentication (VPNGW12:FIA_X509_EXT.2)

2.3.10.1 VPNGW12:FIA_X509_EXT.2.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.3.10.2 VPNGW12:FIA_X509_EXT.2.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: There is no change to the Evaluation Activities specified for this SFR in the NDcPP Supporting Document. The PP-Module modifies this SFR to support its use for IPsec at a minimum. The evaluator shall ensure that all evaluation of this SFR is performed against its use in IPsec communications as well as any other supported usage.

See NDcPP22e: IA_X509_EXT.2. Note the evaluator confirmed that IPsec is included.

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.3.11 X.509 Certificate Requests (NDcPP22e:FIA_X509_EXT.3)

2.3.11.1 NDcPP22e:FIA_X509_EXT.3.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.3.11.2 NDcPP22e:FIA_X509_EXT.3.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.3 of the ST states the TOE includes a Common Name (CN) and SAN:IPv4 in its CSR.

Component Guidance Assurance Activities: The evaluator shall check to ensure that the guidance documentation contains instructions on requesting certificates from a CA, including generation of a Certification 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.

Section 3.8 of the Admin Guide describes setting the common name in the certificate request. Section 3.13 of the Admin Guide identifies the IP address is required.

Component Testing Assurance Activities: 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 request 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.

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 response message, and demonstrate that the function succeeds.

Test 1- The evaluator followed operational guidance to log into the TOE and then generate the CSR. The evaluator then used the openssl command to ensure that the CSR contains the information claimed in the ST.

Test 2 - The evaluator tested that a certificate without an intermediate CA cannot be validated. The evaluator then demonstrated that a valid certificate signed by a CA that the TOE recognized can be successfully added.

This was repeated for both the EUD and RADIUS/syslog VPNs.

2.4 Security management (FMT)

2.4.1 Management of Security Functions Behaviour (NDcPP22e:FMT_MOF.1/Functions)

2.4.1.1 NDcPP22e:FMT_MOF.1.1/Functions

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: For distributed TOEs see chapter 2.4.1.1.

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).

Section 6.4 of the ST states an administrator can configure the external syslog server.

Component Guidance Assurance Activities: For distributed TOEs see chapter 2.4.1.2.

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.

See NDcPP22e: FMT_SMF.1 for a description of administrative functions provided by the TSS.

See NDcPP22e: FAU_GEN.1 AND FAU_STG_EXT.1 for a description of logging behavior.

Section 3.13 of the Admin Guide describes how an admin can configure the TOE to export audit logs to an IPsec protected syslog server.

Component Testing Assurance Activities: 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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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 Security Administrator authentication shall be successful.

Test 1: There are no non-privileged users, however the results of NDcPP22e:FIA_UIA_EXT.1-t3 demonstrate there are no functions available to users prior to login aside from viewing the login banner.

Test 2: This was tested as part of NDcPP22e:FAU_STG_EXT.1. The evaluator modified parameters for configuration of the transmission protocol for transmission of audit data to an external IT entity successfully as an authenticated administrator.

Test 3: This test is not applicable because the selection 'determine the behavior of' was not made in this Security Target.

Test 4: This test is not applicable because the selection 'determine the behavior of' was not made in this Security Target.

2.4.2 Management of security functions behaviour (NDcPP22e:FMT_MOF.1/ManualUpdate)

2.4.2.1 NDcPP22e:FMT_MOF.1.1/ManualUpdate

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

There are no specific requirements for non-distributed TOEs.

There are no specific requirements for this non-distributed TOE.

Component Guidance Assurance Activities: 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).

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).

The “Software Update” section of the Admin Guide explains how to install a software update. It identifies where to get the update and the process for applying the update. The TOE will automatically reboot when the software update is complete and it will be running the newly installed software.

Component Testing Assurance Activities: 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 should fail.

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.

The evaluator attempted commands to perform an update prior to login and confirmed that the configuration could not be viewed or modified prior to authentication.

The evaluator attempted an update after authenticating, see FPT_TUD_EXT.1 for the results of this test.

2.4.3 Management of TSF Data (NDcPP22e:FMT_MTD.1/CoreData)

2.4.3.1 NDcPP22e:FMT_MTD.1.1/CoreData

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

If the 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.

Section 6.4 of the ST states that The TOE provides no access to any services prior to login other than clearing keys (through a physical switch on the TOE's exterior). This prevents non-administrative users from accessing TOE services.

Section 6.4 of the ST explains The TOE allows only security administrators to manage (i.e., import or delete) root CAs (used during IKE certificate authentication) and the certificates the TOE uses to authenticate itself to IKE peers. This represents the TOE's trust store for X.509v3 certificates. The TOE restricts changes to its trust store to authenticated administrators, who can make changes using the Device Provisioning Application (DPA).

Component Guidance Assurance Activities: 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.

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.

The only users permitted to log onto the TOE are administrators, so all such access is limited to administrators.

Section "Administration Functions" describes the non-protocol related management functions.

Sections "Configuring Wired, Wi-Fi, and VPN Data" and "Importing X.509 Certificates" describe configuring protocols and this includes creating certificates and managing the trust store.

Section “Optional Pull-down Menu Items” includes option 5, Import Trusted CA Certs, which states that certs loaded by this option are inserted as a trust anchor. This is the only way to designate a cert as a trust anchor.

Component Testing Assurance Activities: 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.

All of the management functions have been exercised under the other SFRs as demonstrated throughout the AAR.

2.4.4 Management of TSF Data (NDcPP22e:FMT_MTD.1/CryptoKeys)

2.4.4.1 NDcPP22e:FMT_MTD.1.1/CryptoKeys

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: For distributed TOEs see chapter 2.4.1.1.

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.

Section 6.4 of the ST explains the TOE allows only security administrators to manage (i.e., import or delete) root CAs (used during IKE certificate authentication) and the certificates the TOE uses to authenticate itself to IKE peers. The TOE permits the administrator to make such changes using the Device Provisioning Application (DPA).

Component Guidance Assurance Activities: For distributed TOEs see chapter 2.4.1.2.

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.

Sections “Configuring Wired, Wi-Fi, and VPN Data” and “Importing X.509 Certificates” describe configuring protocols and this includes creating certificates and managing the trust store

Section “Key Status / Zeroize” describes the process of zeroizing the TOE which is the only method to delete the Keys on the TOE.

Section “Optional Pull-down menu” mentions the ability to generate new certificate signing request which generates new keys.

The TOE does not support the ability to modify keys.

Component Testing Assurance Activities: 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. The evaluator shall try to perform at least one of the related actions with prior authentication as security administrator. This attempt should be successful.

The successful generation of a new CSR and private key, and the subsequent import of CA and signed certificate by a authorized administrator is demonstrated in NDcPP22e:FIA_X509_EXT.3.

The results of NDcPP22e:FIA_UIA_EXT.1-t3 demonstrate there are no functions available to users prior to login aside from viewing the login banner.

2.4.5 Management of TSF Data (VPNGW12:FMT_MTD.1/CryptoKeys)

2.4.5.1 VPNGW12:FMT_MTD.1.1/CryptoKeys

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: There is no change to the Evaluation Activities specified for this SFR in the NDcPP Supporting Document. The PP-Module modifies this SFR to make it mandatory because of the TOE's required support for IPsec.

See NDCPP22e:FMT_MTD.1.

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.4.6 Specification of Management Functions - per TD0631 (NDcPP22e:FMT_SMF.1)

2.4.6.1 NDcPP22e:FMT_SMF.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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(1)/ManualUpdate, FMT_MOF.1(4)/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(2)/Services, and FMT_MOF.1(3)/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.

(containing also requirements on Guidance Documentation and Tests)

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).

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.

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.

Section 6.4 of the ST states the TOE makes its administrative account management services available through its local (USB) and remote (Ethernet) administrative interfaces. The ST reference the requirement for a list of management functions supported.

The “Account Configuration” section of the Admin Guide explains that all administration is done via SSH. The “Configuring Wired, Wi-Fi, and VPN Data” section identities where the IP address of the wired management interface is entered.

Section 3.1 “Setup” of the Admin Guide states The TOE provides two interfaces a data interface (Ethernet interface through pogo pins) and a local interface (acting as a USB to Ethernet peripheral). An administrator uses the SSH through local interface to provision the TOE and for local access after provisioning. An administrator can use the data interface to export audit logs post-mission (or in alternative configurations, an administrator can use the data interface to support a both connections to a WPA-Enterprise/RADIUS server as well as remote SSH administrative access).

Component Guidance Assurance Activities: See TSS Assurance Activities

Component Testing Assurance Activities: 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.

The TOE is compliant with all requirements in the ST as identified in this report. The evaluator used the Guidance as part of testing and was able to configure all claimed functionality in order to perform the associated tests as identified in the test assurance activities.

2.4.7 Specification of Management Functions (WLAN Access Systems) (WLANAS10:FMT_SMF.1/AccessSystem)

2.4.7.1 WLANAS10:FMT_SMF.1.1/AccessSystem

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator will confirm that the TSS includes which security types (e.g., WPA3), authentication protocol (e.g., SAE), and frequency bands the WLAN AS supports. The evaluator will confirm that the TSS includes how connection attempts from clients that are not operating on an approved security type are handled.

Section 6.4 of the ST states the TOE supports WPA2-Enterprise mode and support both WPA3-Enterprise and WPA-SAE modes in 2.4 and 5 GHz radio frequencies. The TOE rejects any connections from clients that do not offer a security type matching those the TOE provides. The TOE allows an administrator to use the Device Provisioning Application (DPA) to configure the Security type (WPA2 or WPA3), Authentication protocol (SAE or Enterprise), the SSID, the Transmit power level, and the client authentication credential (SAE passphrase).

Component Guidance Assurance Activities: The evaluator will confirm that the operational guidance includes instructions for configuring the WLAN AS for each feature listed.

The “Configuring Wired, Wi-Fi, and VPN Data” section of the Admin Guide identifies the following settings. All in the requirement ar eaddressed:

- SSID is hidden – check this box if you don’t want the SSID advertised.
- SSID – The name advertised by the TOE for Wi-Fi connections.
- IP Address – Address of the TOE’s Wi-Fi interface and it must end in “.1”. EUDs will be given DHCP address of “.2”, “.3”, etc.
- Num Devices – The max number of EUDs to be connected to this RAP.
- 80211 Mode – The mode and channel bandwidth of the Wi-Fi configuration.
- Channel – The Wi-Fi channel to be used.
- Encryption – The Wi-Fi encryption to be used. Note: Using any of the CCMP-256 settings may result in a reduction of about 50% of the Wi-Fi throughput. Some EUDs do not support CCMP-256 mode only.
- Transmit Power – The available transmit power settings (in 1dBm steps from min to max) based on the 80211 Mode selected above.
- WPA3-SAE-PK only mode – check this box if the RAP should only accept Wi-Fi connections in WPA3-SAE-PK mode. This mode is enabled by default and is required for a CSfC compliant configuration. It can only be changed if the CSfC Compliant Mode checkbox under Menu is unchecked

Component Testing Assurance Activities: Test 1: For each security type specified in the TSS, configure the network to the approved security type and verify that the client can establish a connection. Maintaining the same SSID, change the security type of the client to a non-approved security type and attempt to establish a connection. Verify that the connection was unsuccessful.

Test 2: For each authentication protocol specified in the TSS, configure the network accordingly per the AGD. Verify that the client connection attempt is successful when using the correct client credentials and that the connection is unsuccessful when incorrect authentication credentials are used.

Test 3: Configure the SSID to be broadcasted. Using a network sniffing tool, capture a beacon frame and confirm that the SSID is included. Configure the SSID to be hidden. Using a network sniffing tool, capture a beacon frame and confirm that the SSID is not listed.

Test 4: The evaluator will configure the AS to operate in each of the selected frequency bands and verify using a network sniffing tool.

Test 5: The evaluator will demonstrate that the client can establish a connection to the AS on the default power level. After disconnecting, the power level should be adjusted and then the client should be able to successfully connect to the AS again.

Test 1: For each security type claimed by the TOE, the evaluator first configured the WLAN client to use the approved security type and attempted a connection The evaluator observed these connection attempts were successful. The evaluator then maintained the SSID configured on the TOE, and configured the client for a security type that was not configured/approved on the TOE. The evaluator then caused the client to attempt a connection to the TOE. The evaluator observed that these connection attempts using a non-approved security type were unsuccessful.

Test 2: The evaluator attempted two connections with each authentication protocol (WPA3-SAE, WPA2 Enterprise, WPA3 Enterprise). The first attempt had the client provide correct authentication credentials, and the connection was successful. The following attempt had the client provide incorrect credentials, and the connection was unsuccessful.

Test 3: The evaluator configured the TOE to broadcast its SSID following Operational Guidance. The evaluator started a sniffing session of wireless traffic, and confirmed that the resulting packet capture included a beacon frame that listed the TOEs SSID. The evaluator then configured the TOE to hide its SSID, and repeated the process. The evaluator confirmed that the resulting packet capture included a beacon frame that did not list the SSID.

Test 4: The evaluator configured the TOE following operational guidance to broadcast only on each of the claimed bands (2.4ghz, 5ghz) For each band, the evaluator disabled the other two and attempted to connect a client, which was successful. The evaluator then verified that the client connected with the configured band using a command on the TOE.

Test 5: For each of the claimed radio band frequencies (2.4ghz, 5ghz), the evaluator first demonstrated a successful connection from a client to the AS using the default power level. The evaluator then disconnected from the AS, then adjusted the power level. With this new power level setting, the evaluator attempted to connect the client again and observed that the connection attempt was successful.

2.4.8 Specification of Management Functions (VPNGW12:FMT_SMF.1/VPN)

2.4.8.1 VPNGW12:FMT_SMF.1.1/VPN

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to confirm that all management functions specified in FMT_SMF.1/VPN are provided by the TOE. As with FMT_SMF.1 in the Base-PP, the evaluator shall ensure that the TSS identifies what logical interfaces are used to perform these functions and that this includes a description of the local administrative interface.

Section 6.4 of the ST state the TOE provides all management functions listed in VPNGW12:FMT_SMF.1/VPN through both its local (USB) and remote (Ethernet) management interfaces. Those include definition of packet filtering rules, associate of those rules to network interfaces, and ordering of those rules by priority.

Component Guidance Assurance Activities: The evaluator shall examine the operational guidance to confirm that all management functions specified in FMT_SMF.1/VPN are provided by the TOE. As with FMT_SMF.1 in the Base-PP, the evaluator shall ensure that the operational guidance identifies what logical interfaces are used to perform these functions and that this includes a description of the local administrative interface.

Section 3.1 "Setup" of the Admin Guide states The TOE provides two interfaces a data interface (Ethernet interface through pogo pins) and a local interface (acting as a USB to Ethernet peripheral). An administrator uses the SSH through local interface to provision the TOE and for local access after provisioning. An administrator can use the data interface to export audit logs post-mission (or in alternative configurations, an administrator can use the data interface to support a both connections to a WPA-Enterprise/RADIUS server as well as remote SSH administrative access).

Component Testing Assurance Activities: The evaluator tests management functions as part of performing other test EAs. No separate testing for FMT_SMF.1/VPN is required unless one of the management functions in FMT_SMF.1.1/VPN has not already been exercised under any other SFR.

All TOE security functions are identified in the Guidance and have been tested as documented throughout this AAR.

2.4.9 Restrictions on Security Roles (NDcPP22e:FMT_SMR.2)

2.4.9.1 NDcPP22e:FMT_SMR.2.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.4.9.2 NDcPP22e:FMT_SMR.2.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.4.9.3 NDcPP22e:FMT_SMR.2.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.4 of the ST states the TOE supports only local and remote administrator and places no restrictions on these roles other than exempting local administration from incorrect password lock out

Component Guidance Assurance Activities: 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.

Section 3.1 “Setup” of the Admin Guide states The TOE provides two interfaces a data interface (Ethernet interface through pogo pins) and a local interface (acting as a USB to Ethernet peripheral). An administrator uses the SSH through local interface to provision the TOE and for local access after provisioning. An administrator can use the data interface to export audit logs post-mission (or in alternative configurations, an administrator can use the data interface to support a both connections to a WPA-Enterprise/RADIUS server as well as remote SSH administrative access).

Component Testing Assurance Activities: 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.

The TOE only supports SSH as an administrative interface and was tested throughout the course of testing.

2.4.10 No Administration from Client (WLANAS10:FMT_SMR_EXT.1)

2.4.10.1 WLANAS10:FMT_SMR_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: The evaluator will 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. The evaluator will confirm that the TOE does not permit remote administration from a wireless client by default.

Section 3.1 “Setup” section of the Admin Guide states The TOE provides two interfaces a data interface (Ethernet interface through pogo pins) and a local interface (acting as a USB to Ethernet peripheral). An administrator uses the SSH through local interface to provision the TOE and for local access after provisioning. An administrator can use the data interface to export audit logs post-mission (or in alternative configurations, an administrator can use the data interface to support a both connections to a WPA-Enterprise/RADIUS server as well as remote SSH administrative access).

Section 3.2 “Account Configuration” of the Admin Guide states TOE administrators manage the security functions of the TOE through the Secure Shell Protocol (SSH) CLI. Administration cannot be performed from a wireless client.

Component Testing Assurance Activities: The evaluator will 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 will then demonstrate that an identically configured wireless client that can successfully connect to the TOE cannot be used to perform administration.

Test 1: The evaluator first configured a device with a wired connection to the TOE, successfully established an SSH connection to the TOE, and performed administrative actions. Next the evaluator attempted the same connections from a wireless client connected to the TOE’s wireless (Wi-Fi) network and found that the evaluator could not successfully establish an SSH connection to the TOE.

2.5 Packet Filtering (FPF)

2.5.1 Packet Filtering Rules - per TD0683 (VPNGW12:FPF_RUL_EXT.1)

2.5.1.1 VPNGW12:FPF_RUL_EXT.1.1

TSS Assurance Activities: The evaluator shall verify that the TSS provide a description of the TOE's initialization/startup process, which clearly indicates where processing of network packets begins to take place, and provides a discussion that supports the assertion that packets cannot flow during this process.

The evaluator shall verify that the TSS also includes a narrative that identifies the components (e.g., active entity such as a process or task) involved in processing the network packets and describes the safeguards that would prevent packets flowing through the TOE without applying the ruleset in the event of a component failure. This could include the failure of a component, such as a process being terminated, or a failure within a component, such as memory buffers full and cannot process packets.

Section 6.5 of the ST states the TOE's boot process brings up firewall rules (netfilter) prior to bringing up networking interfaces, and as such cannot send unfiltered traffic. The TOE's netfilter kernel process bears responsibility for processing network packets and processes each packet against the netfilter rules governing each input and output chain. Should netfilter fail, because it executes within the kernel, its failure would trigger a kernel panic and result in the TOE restarting.

Guidance Assurance Activities: The operational guidance associated with this requirement is assessed in the subsequent test EAs.

Refer to the subsequent test assurance activities below.

Testing Assurance Activities: Test 1: The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would otherwise be denied by the ruleset should be sourced and directed to a host. The evaluator shall use a packet sniffer to verify none of the generated network traffic is permitted through the TOE during initialization.

Test 2: The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would be permitted by the ruleset should be sourced and directed to a host. The evaluator shall use a packet sniffer to verify none of the generated network traffic is permitted through the TOE during initialization and is only permitted once initialization is complete.

Note: The remaining testing associated with application of the ruleset is addressed in the subsequent test Evaluation Activities.

Test 1: The evaluator configured TOE packet filter rules to block specific ICMPv4 and ICMPv6 traffic, then generated ICMP traffic matching the rules, directed that traffic to the TOE, and rebooted to the TOE. The evaluator observed that the TOE correctly blocked the ICMP traffic at all times including before, during, and after reboot.

Test 2: The evaluator configured TOE packet filter rules to allow a different set of ICMPv4 and ICMPv6 traffic, then generated ICMP traffic matching the rules, directed that ICMP traffic to the Toe, and rebooted the TOE. The evaluator observed that the TOE correctly permitted the ICMP traffic after initialization, but did not permit the ICMP traffic during its reboot.

2.5.1.2 VPNGW12:FPF_RUL_EXT.1.2

TSS Assurance Activities: There are no EAs specified for this element. Definition of packet filtering policy, association of operations with packet filtering rules, and association of these rules to network interfaces is described collectively under FPF_RUL_EXT.1.4.

See VPNGW12: FPF_RUL_EXT.1.4

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.5.1.3 VPNGW12:FPF_RUL_EXT.1.3

TSS Assurance Activities: There are no EAs specified for this element. Definition of packet filtering policy, association of operations with packet filtering rules, and association of these rules to network interfaces is described collectively under FPF_RUL_EXT.1.4.

See VPNGW12: FPF_RUL_EXT.1.4

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.5.1.4 VPNGW12:FPF_RUL_EXT.1.4

TSS Assurance Activities: The evaluator shall verify that the TSS describes a packet filtering policy that can use the following fields for each identified protocol, and that the RFCs identified for each protocol are supported:

- IPv4 (RFC 791)

o source address

o destination address

o Protocol

- IPv6 (RFC 8200)

o source address

o destination address

o next header (protocol)

- TCP (RFC 793)

o source port

o destination port

- UDP (RFC 768)

o source port

o destination port

The evaluator shall verify that the TSS describes how conformance with the identified RFCs has been determined by the TOE developer (e.g., third party interoperability testing, protocol compliance testing).

The evaluator shall verify that each rule can identify the following actions: permit, discard, and log.

The evaluator shall verify that the TSS identifies all interface types subject to the Packet Filtering policy and explains how rules are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used), they can be treated collectively as a distinct network interface.

Section 6.5 of the ST states the TOE implements a packet filtering policy implementation that can use the following fields in these RFC protocols:

The TSF shall allow the definition of Packet Filtering rules (permit, discard, and log) using the following network protocols and protocol fields:

- IPv4 (RFC 791)

  o source address

  o destination address

  o protocol

- IPv6 (RFC 2460)

  o source address

  o destination address

  o next Header (protocol)

- TCP (RFC 793)

  o source port

o destination port

- UDP (RFC 768)

o source port

o destination port.

The TOE only has a single Ethernet interface (beyond its wireless and USB interfaces). During development, the vendor determined compliance by examining the TOE's open-source implementation to ensure compliance and by also performing limited independent testing to ensure that the implementation could correctly filter based upon the above protocols and protocol fields.

Guidance Assurance Activities: The evaluators shall verify that the operational guidance identifies the following protocols as being supported and the following attributes as being configurable within packet filtering rules for the associated protocols:

- IPv4 (RFC 791)

o source address

o destination address

o Protocol

- IPv6 (RFC 8200)

o source address

o destination address

o next header (protocol)

- TCP (RFC 793)

o source port

o destination port

- UDP (RFC 768)

o source port

o destination port

The evaluator shall verify that the operational guidance indicates that each rule can identify the following actions: permit, discard, and log.

The evaluator shall verify that the operational guidance explains how rules are associated with distinct network interfaces.

The guidance may describe the other protocols contained within the ST (e.g., IPsec, IKE, potentially HTTPS, SSH, and TLS) that are processed by the TOE. The evaluator shall ensure that it is made clear what protocols were not considered as part of the TOE evaluation.

Section 3.14 "IP Filtering Configuration and SPD rules" of the Admin Guide includes Table 2, identifying the protocols, how to configure permit discard, and log actions, and how to associate rules with a distinct network interface.

Testing Assurance Activities: The evaluator shall perform the following tests:

Test 1: The evaluator shall use the instructions in the operational guidance to test that packet filter rules can be created that permit, discard, and log packets for each of the following attributes:

- IPv4

o Source address

o Destination Address

o Protocol

- IPv6

o Source Address

o Destination Address

o Next Header (Protocol)

- TCP

o Source Port

o Destination Port

- UDP

o Source Port

o Destination Port

Test 2: The evaluator shall repeat Test 1 above for each distinct network interface type supported by the TOE to ensure that Packet filtering rules can be defined for each all supported types.

Note that these test activities should be performed in conjunction with those of FPF_RUL_EXT.1.6 where the effectiveness of the rules is tested; here the evaluator is just ensuring the guidance is sufficient and the TOE supports the administrator creating a ruleset based on the above attributes. The test activities for FPF_RUL_EXT.1.6 define the protocol/attribute combinations required to be tested. If those combinations are configured manually, that will fulfill the objective of these test activities, but if those combinations are configured otherwise (e.g., using automation), these test activities may be necessary in order to ensure the guidance is correct and the full range of configurations can be achieved by a TOE administrator.

Test 1 & 2: The evaluator configured firewall rules for testing of the other VPNGW12:FPF_RUL_EXT.1 (including VPNGW12:FPF_RUL_EXT.1.6) tests using instructions provided within the administrative guidance and found all necessary instructions were provided accurately. The tests performed for FPF_RUL_EXT.1.6 incorporate numerous variations of packet filtering rules that demonstrate proper enforcement of the packet filtering ruleset (e.g., permit, deny, and log rules, IPv4 and IPv6, TCP and UDP, numerous ports, source & destination differences, and transport protocols).

2.5.1.5 VPNGW12:FPF_RUL_EXT.1.5

TSS Assurance Activities: The evaluator shall verify that the TSS describes the algorithm applied to incoming packets, including the processing of default rules, determination of whether a packet is part of an established session, and application of administrator defined and ordered ruleset.

Section 6.5 of the ST states the TOE processes incoming packets in netfilter's chain applying the administrator defined rulesets in order. The TOE also includes a set of default rules, which restrict the TOE to the minimum necessary traffic needed for the WLAN (802.11 Wi-Fi) IPsec tunnel, SSH administration of the RAP, and the RAP's outgoing connections for syslog and RADIUS. The TOE also inspects packets to determine whether those packets are part of an established session, and if so, applies the administrator defined rulesets accordingly.

Guidance Assurance Activities: The evaluator shall verify that the operational guidance describes how the order of Packet filtering rules is determined and provides the necessary instructions so that an administrator can configure the order of rule processing.

Section 3.14 "IP Filtering Configuration and SPD rules" of the Admin Guide states The IP Filtering tab of DPA allows the admin to create one or more of IP filter and apply those rules to the TOE. Each rule can be specified by configuring the options from Table 2. Once the options are configured, the "Add Rule to Scratchpad" button is pressed and the rule will show up in the Rules Scratchpad. Rules can be deleted from the Rules Scratchpad by specifying the "Rule Num" pull-down menu and then selecting the "Delete Rule from Scratchpad" button. The "Clear Scratchpad" button will clear the Rules Scratchpad of all rules. Once a satisfactory set of rules is defined, the "Apply Rules to RAP" button is pressed to configure the RAP with this set of rules. The RAP enforces the rules, prioritizing earlier rules over later rules. The RAP's IP filtering can be reset to the default set of rules by pressing the "Reset RAP to Default Rules" button.

Testing Assurance Activities: The evaluator shall perform the following tests:

Test 1: The evaluator shall devise two equal Packet Filtering rules with alternate operations - permit and discard. The rules should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first rule is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation.

Test 2: The evaluator shall repeat the procedure above, except that the two rules should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.

Test 1: The evaluator configured the TOE (according to the admin guide) with two packet filtering rules using the same matching criteria, where one rule would permit while the other would deny traffic. Packets matching the ACL entry rule were sent through the TOE and the evaluator observed that the action taken by the TOE matched the action specified by the first ACL entry.

Test 2: Continuing test 1, the evaluator repeated the procedure above, except the evaluator changed the rules to make one a subset of the other, and then tested both orders. The evaluator confirmed that the first rule is enforced regardless of the specificity of the rule.

2.5.1.6 VPNGW12:FPF_RUL_EXT.1.6

TSS Assurance Activities: The evaluator shall verify that the TSS describes the process for applying Packet Filtering rules and also that the behavior (either by default, or as configured by the administrator) is to discard packets when there is no rule match. The evaluator shall verify the TSS describes when the IPv4/IPv6 protocols supported by the TOE differ from the full list provided in the RFC Values for IPv4 and IPv6 table.

Section 6.5 of the ST states the TOE has a default discard rule which drops packets that match no existing or administrator defined rule, and the TOE's supports the full list of IPv4/IPv6 protocols and does not differ.

Guidance Assurance Activities: The evaluator shall verify that the operational guidance describes the behavior if no rules or special conditions apply to the network traffic. If the behavior is configurable, the evaluator shall verify that the operational guidance provides the appropriate instructions to configure the behavior to discard packets with no matching rules. The evaluator shall verify that the operational guidance describes the range of IPv4 and IPv6 protocols supported by the TOE.

Section 3.14 "IP Filtering Configuration and SPD rules" of the Admin Guide states The TOE, by default, drops all input and output packets, if no rule exists to specifically allow that traffic.

Testing Assurance Activities: The evaluator shall perform the following tests:

Test 1: The evaluator shall configure the TOE to permit and log each supported IPv4 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each supported IPv4 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that the supported protocols are permitted (i.e., by capturing the packets after passing through the TOE) and logged. Any protocols not supported by the TOE must be denied.

Test 2: The evaluator shall configure the TOE to permit all traffic except to discard and log each supported IPv4 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv4 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that the supported protocols are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Any protocols not supported by the TOE must also be denied but are not required to be logged.

Test 3: The evaluator shall configure the TOE to permit and log each supported IPv4 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. Additionally, the evaluator shall configure the TOE to discard and log each supported IPv4 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with different (than those permitted above) combinations of a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each supported IPv4 Transport Layer Protocol and outside the scope of all source and destination addresses configured above in order to ensure that the supported protocols are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Any protocols not supported by the TOE must be denied.

Test 4: The evaluator shall configure the TOE to permit and log each supported IPv6 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that the supported protocols are permitted (i.e., by capturing the packets after passing through the TOE) and logged. Any protocols not supported by the TOE must be denied.

Test 5: The evaluator shall configure the TOE to permit all traffic except to discard and log each supported IPv6 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that the supported protocols are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Any protocols not supported by the TOE must also be denied but are not required to be logged.

Test 6: The evaluator shall configure the TOE to permit and log each supported IPv6 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. Additionally, the evaluator shall configure the TOE to discard and log each supported IPv6 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with different (than those permitted above) combinations of a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and outside the scope of all source and destination addresses configured above in order to ensure that the supported protocols are dropped (i.e., by capturing no applicable packets passing through the TOE) and logged. Any protocols not supported by the TOE must be denied.

Test 7: The evaluator shall configure the TOE to permit and log protocol 6 (TCP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination TCP ports in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.

Test 8: The evaluator shall configure the TOE to discard and log protocol 6 (TCP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination TCP ports in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.

Test 9: The evaluator shall configure the TOE to permit and log protocol 17 (UDP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination UDP ports in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged. Here the evaluator ensures that the UDP port 500 (IKE) is included in the set of tests.

Test 10: The evaluator shall configure the TOE to discard and log protocol 17 (UDP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination UDP ports in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Again, the evaluator ensures that UDP port 500 is included in the set of tests.

The following table identifies the RFC defined values for the protocol fields for IPv4 and IPv6 to be used in configuring and otherwise testing Packet Filtering rule definition and enforcement:

Test 1: The evaluator attempted to send packets through the TOE when the TOE had the following rules configured. The packets that were sent were constructed such that the packet would match only one configured rule. The evaluator confirmed that packets sent using protocols that are not supported by the TOE are denied.

- IPv4 Protocol permitted and logged based on specific source and destination addresses.

• IPv4 Protocol permitted and logged based on specific destination addresses.

• IPv4 Protocol permitted and logged based on specific source addresses.

• IPv4 Protocol permitted and logged based on wildcard addresses.

Test 2: The evaluator attempted to send packets through the TOE when the TOE had the following rules configured: The packets that were sent were constructed such that the packet would match only one configured rule. The evaluator confirmed that packets sent using protocols that are not supported by the TOE are denied.

• All traffic permitted except to deny and log each defined IPv4 Transport Layer Protocol in conjunction with a specific source address and specific destination address.

• All traffic permitted except to deny and log each defined IPv4 Transport Layer Protocol in conjunction with a specific destination address.

• All traffic permitted except to deny and log each defined IPv4 Transport Layer Protocol in conjunction with a specific source address.

• All traffic permitted except to deny and log each defined IPv4 Transport Layer Protocol in conjunction with wildcard source and destination addresses.

Test 3: The evaluator attempted to send packets through the TOE when the TOE had the following rules configured: The packets that were sent were constructed such that the packet would match only one configured rule. The evaluator confirmed that packets sent using protocols that are not supported by the TOE are denied.

• IPv4 Protocol some permitted and logged and some denied and logged based on specific source and destination addresses.

• IPv4 Protocol some permitted and logged and some denied and logged based on specific destination addresses.

IPv4 Protocol some permitted and logged and some denied and logged based on specific source addresses.

• IPv4 Protocol some permitted and logged and some denied and logged based on wildcard addresses.

• IPv4 Protocol some permitted and logged and some denied and logged based on default addresses.

Test 4: The evaluator attempted to send packets through the TOE when the TOE had the following rules configured: The packets that were sent were constructed such that the packet would match only one configured rule. The evaluator confirmed that packets sent using protocols that are not supported by the TOE are denied.

Additionally, supported protocols are filtered properly and the rules are enforced.

• IPv6 Protocol permitted and logged based on specific source and destination addresses.

• IPv6 Protocol permitted and logged based on specific destination addresses.

• IPv6 Protocol permitted and logged based on specific source addresses.

• IPv6 Protocol permitted and logged based on wildcard addresses.

Test 5: The evaluator attempted to send packets through the TOE when the TOE had the following rules configured: The packets that were sent were constructed such that the packet would match only one configured rule. The evaluator confirmed that packets sent using protocols that are not supported by the TOE are denied.

Additionally, supported protocols are filtered properly and the rules are enforced.

• IPv6 Protocol all permitted with some denied and logged based on specific source and destination

addresses.

• IPv6 Protocol all permitted with some denied and logged based on specific destination addresses.

• IPv6 Protocol all permitted with some denied and logged based on specific source addresses.

• IPv6 Protocol all permitted with some denied and logged based on wildcard addresses.

Test 6: The evaluator attempted to send packets through the TOE when the TOE had the following rules configured: The packets that were sent were constructed such that the packet would match only one configured rule. The evaluator confirmed that packets sent using protocols that are not supported by the TOE are denied.

Additionally, supported protocols are filtered properly and the rules are enforced.

• IPv6 Protocol some permitted and logged and some denied and logged based on specific source and destination addresses.

• IPv6 Protocol some permitted and logged and some denied and logged based on specific destination addresses.

• IPv6 Protocol some permitted and logged and some denied and logged based on specific source addresses.

• IPv6 Protocol some permitted and logged and some denied and logged based on wildcard addresses.

• IPv6 Protocol some permitted and logged and some denied and logged based on default addresses.

Test 7: The evaluator attempted to send packets through the TOE when the TOE had the following rules configured: The packets that were sent were constructed such that the packet would match only one configured rule. The evaluator confirmed that packets sent using protocols that are not supported by the TOE are denied.

Additionally, supported protocols are filtered properly and the rules are enforced.

• TCP (IPv4) permitted and logged based on source port.

• TCP (IPv4) permitted and logged based on destination port.

- TCP (IPv4) permitted and logged based on source and destination port.

- TCP (IPv6) permitted and logged based on source port.

- TCP (IPv6) permitted and logged based on destination port.

- TCP (IPv6) permitted and logged based on source and destination port.

Test 8: The evaluator attempted to send packets through the TOE when the TOE had the following rules

configured: The packets that were sent were constructed such that the packet would match only one configured rule. The evaluator confirmed that packets sent using protocols that are not supported by the TOE are denied.

Additionally, supported protocols are filtered properly and the rules are enforced.

- TCP (IPv4) denied and logged based on source port.

- TCP (IPv4) denied and logged based on destination port.

- TCP (IPv4) denied and logged based on source and destination port.

- TCP (IPv6) denied and logged based on source port.

- TCP (IPv6) denied and logged based on destination port.

- TCP (IPv6) denied and logged based on source and destination port.

Test 9: The evaluator attempted to send packets through the TOE when the TOE had the following rules

configured: The packets that were sent were constructed such that the packet would match only one configured rule. The evaluator confirmed that packets sent using protocols that are not supported by the TOE are denied.

Additionally, supported protocols are filtered properly and the rules are enforced.

- UDP (IPv4) permitted and logged based on source port.

- UDP (IPv4) permitted and logged based on destination port.

- UDP (IPv4) permitted and logged based on source and destination port.

- UDP (IPv6) permitted and logged based on source port.

- UDP (IPv6) permitted and logged based on destination port.

- UDP (IPv6) permitted and logged based on source and destination port.

Test 10: The evaluator attempted to send packets through the TOE when the TOE had the following rules configured: The packets that were sent were constructed such that the packet would match only one configured rule. The evaluator confirmed that packets sent using protocols that are not supported by the TOE are denied.

Additionally, supported protocols are filtered properly and the rules are enforced.

- UDP (IPv4) denied and logged based on source port.
- UDP (IPv4) denied and logged based on destination port.
- UDP (IPv4) denied and logged based on source and destination port.
- UDP (IPv6) denied and logged based on source port.
- UDP (IPv6) denied and logged based on destination port.
- UDP (IPv6) denied and logged based on source and destination port.

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.6 Protection of the TSF (FPT)

2.6.1 Protection of Administrator Passwords (NDcPP22e:FPT_APW_EXT.1)

2.6.1.1 NDcPP22e:FPT_APW_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.6.1.2 NDcPP22e:FPT_APW_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.6 of the ST states the TOE stores administrative user SSH passwords hashed with SHA-256.

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.6.2 Failure with Preservation of Secure State (WLANAS10:FPT_FLS.1)

2.6.2.1 WLANAS10:FPT_FLS.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator will examine the TSS to determine that the TOE's implementation of the fail secure functionality is documented. The evaluator will examine the TSS to ensure that it describes all failure conditions and how a secure state is preserved if any of these failures occur. The evaluator will ensure that the definition of a secure state is suitable to ensure the continued protection of any key material and user data.

Section 6.6 of the ST states that if the TOE fails a power-up self-test, fails noise source health tests during boot, or fails the image integrity check, the TOE will prevent further execution by halting its boot process, making key material inaccessible.

Component Guidance Assurance Activities: The evaluator will examine the operational guidance to verify that it describes applicable recovery instructions for each TSF failure state.

Section 3.17 “Powering the TOE in Operational Settings” of the Admin Guide states In operational settings, the TOE is powered by a PRC-117G radio or a mounting fixture. As soon as normal power is applied to the TOE, it reboots and lights up the yellow LED. During this time, the TOE performs power up self-tests to ensure the integrity of its firmware and to ensure correct operation of its cryptographic algorithms. If a self-test fails, the TOE flashes it LEDs and halts its boot. An administrator can attempt to power cycle the TOE to see if it can boot normally, otherwise, the unit must be returned to DataSoft. If all self-tests pass, the TOE indicates this by turning off the yellow LED at which point the TOE becomes fully operational.

Component Testing Assurance Activities: For each failure mode specified in the ST, the evaluator will ensure that the TOE attains a secure state (e.g., shutdown) after initiating each failure mode type.

The evaluator verified that for each failure mode specified in the ST, that the TOE attains a secure state (shutdown) after initiating each failure mode type.

2.6.3 Failure with Preservation of Secure State (Self-Test Failures) (VPNGW12:FPT_FLS.1/SelfTest)

2.6.3.1 VPNGW12:FPT_FLS.1.1/SelfTest

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall ensure the TSS describes how the TOE ensures a shutdown upon a self-test failure, a failed integrity check of the TSF executable image, or a failed health test of the noise source. If there are instances when a shut-down does not occur, (e.g., a failure is deemed non- security relevant), the evaluator shall ensure that those cases are identified and a rationale is provided that supports the classification and justifies why the TOE's ability to enforce its security policies is not affected in any such instance.

Section 6.6 of the ST states that if the TOE fails a power-up self-test, fails noise source health tests during boot, or fails the image integrity check, the TOE will prevent further execution by halting its boot process, making key material inaccessible.

Component Guidance Assurance Activities: The evaluator shall verify that the operational guidance provides information on the self-test failures that can cause the TOE to shut down and how to diagnose the specific failure that has occurred, including possible remediation steps if available.

Section 3.17 “Powering the TOE in Operational Settings” of the Admin Guide states In operational settings, the TOE is powered by a PRC-117G radio or a mounting fixture. As soon as normal power is applied to the TOE, it reboots and lights up the yellow LED. During this time, the TOE performs power up self-tests to ensure the integrity of its firmware and to ensure correct operation of its cryptographic algorithms. If a self-test fails, the TOE flashes it LEDs and halts its boot. An administrator can attempt to power cycle the TOE to see if it can boot normally, otherwise, the unit must be returned to DataSoft. If all self-tests pass, the TOE indicates this by turning off the yellow LED at which point the TOE becomes fully operational.

Component Testing Assurance Activities: None Defined

2.6.4 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) (NDcPP22e:FPT_SKP_EXT.1)

2.6.4.1 NDcPP22e:FPT_SKP_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.6 of the ST states the TOE stores IKE authentication certificates and SSH public/private keys and while the TOE provides administrative access to update or replace these keys, the TOE provides no method to view or output these values (even to authorized administrators). The TOE stores all keys in keys plaintext within the TOE's internal filesystem, to which administrators have no access.

As identified in Section 6.2, all values are stored in plaintext.

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.6.5 Reliable Time Stamps - per TD0632 (NDcPP22e:FPT_STM_EXT.1)

2.6.5.1 NDcPP22e:FPT_STM_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.6.5.2 NDcPP22e:FPT_STM_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.6 of the ST states the TOE makes use of time when generating audit records (which include a timestamp) and when performing IKE certificate validation (checking certificate validity), and the TOE contains a Real-Time Clock (RTC). After the administrator sets the TOE's clock during provisioning, the TOE can maintain accurate time using its internal RTC.

Component Guidance Assurance Activities: 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.

In the "Provisioning Application" section of the Admin Guide it explains that the time on the TOE can be updated to match the time on the provisioning laptop by pressing the Update button. The time will also automatically be sent to the TOE during the provisioning step.

Component Testing Assurance Activities: 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.

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.

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.

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.

Test 1 - The evaluator set the local clock from the Device Provisioning Application (DPA) and observed the time change and corresponding audit record.

Test 2 - Not applicable, as the TOE does not claim NTP.

Test 3 - Not applicable, as the TOE does not obtain its time from an underlying VS

2.6.6 TSF TESTING (NDcPP22e:FPT_TST_EXT.1)

2.6.6.1 NDcPP22e:FPT_TST_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

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.

Section 6.6 of the ST states the TOE runs a set of self-test for both its Kernel Cryptography and its OpenSSL library. The Kernel Cryptography self-tests cover AES and the OpenSSL library covers its AES, SHA hashing, HMAC, ECDSA, and DRBG algorithms. In each case, the self-test starts with known data (e.g., a known plaintext, key, and resulting ciphertext) and uses the data to ensure the algorithm works correctly. In addition to these tests, the TOE also ensures the integrity of its firmware image. The TOE's bootloader verifies the ECDSA signature on the main firmware image before uncompressing and executing it. Finally, the TOE's noise source includes a health test to detect if the quality of the noise source degrades, and if so, halts outputting noise.

These tests together ensure that the TOE continues to operate correctly.

Component Guidance Assurance Activities: 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.

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.

Section 3.17 "Powering the TOE in Operational Settings" of the Admin Guide states In operational settings, the TOE is powered by a PRC-117G radio or a mounting fixture. As soon as normal power is applied to the TOE, it reboots and lights up the yellow LED. During this time, the TOE performs power up self-tests to ensure the integrity of its firmware and to ensure correct operation of its cryptographic algorithms. If a self-test fails, the TOE flashes it LEDs and halts its boot. An administrator can attempt to power cycle the TOE to see if it can boot normally, otherwise, the unit must be returned to DataSoft. If all self-tests pass, the TOE indicates this by turning off the yellow LED at which point the TOE becomes fully operational.

Component Testing Assurance Activities: 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 fulfill any of the SFRs.

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.

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.

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.

The TOE performs an integrity check on its image and executes FIPS Power on self-tests during boot. The evaluator booted the TOE and reviewed the results of the self-tests in the console messages.

2.6.7 TSF Testing (VPNGW12:FPT_TST_EXT.1)

2.6.7.1 VPNGW12:FPT_TST_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: There is no change to the Evaluation Activities specified for this SFR in the NDcPP Supporting Document. The PP-Module requires a particular self-test to be performed, but this self-test is still evaluated using the same methods specified in the Supporting Document.

See NDcPP22e:FPT_TST_EXT.1

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.6.8 TSF Testing (WLANAS10:FPT_TST_EXT.1)

2.6.8.1 WLANAS10:FPT_TST_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator will perform the following activities in addition to those required by the NDcPP:

The evaluator will 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 will 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.

The evaluator will also ensure that the TSS or operational guidance describes the actions that take place for successful and unsuccessful execution of the integrity test.

Section 6.6 of the ST states the TOE also ensures the integrity of its firmware image. The TOE's bootloader verifies the ECDSA signature on the main firmware image before uncompressing and executing it. This happens at boot and no administrator action is required..

Component Guidance Assurance Activities: The evaluator will perform the following activities in addition to those required by the NDcPP:

The evaluator will ensure that the TSS or operational guidance describes the actions that take place for successful and unsuccessful execution of the integrity test.

Section 3.17 "Powering the TOE in Operational Settings" of the Admin Guide states In operational settings, the TOE is powered by a PRC-117G radio or a mounting fixture. As soon as normal power is applied to the TOE, it reboots and lights up the yellow LED. During this time, the TOE performs power up self-tests to ensure the integrity of its firmware and to ensure correct operation of its cryptographic algorithms. If a self-test fails, the TOE flashes it LEDs and halts its boot. An administrator can attempt to power cycle the TOE to see if it can boot normally, otherwise, the unit must be returned to DataSoft. If all self-tests pass, the TOE indicates this by turning off the yellow LED at which point the TOE becomes fully operational.

Component Testing Assurance Activities: The evaluator will perform the following activities in addition to those required by the NDcPP:

The evaluator will perform the following tests:

Test 1: Following the operational guidance, the evaluator will initialize the integrity protection system. The evaluator will perform actions to cause TSF software to load and observe that the integrity mechanism does not flag any executables as containing integrity errors.

Test 2: The evaluator will modify the TSF executable and cause that executable to be loaded by the TSF. The evaluator will 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).

Test 1: The evaluator rebooted the TOE and observed that the integrity mechanism did not flag any executables as containing integrity errors.

Test 2: The evaluator attempted to boot an image with an integrity violation and verified that an integrity violation was triggered and the failed integrity image was rejected.

2.6.9 Self-Test with Defined Methods (VPNGW12:FPT_TST_EXT.3)

2.6.9.1 VPNGW12:FPT_TST_EXT.3.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.6.9.2 VPNGW12:FPT_TST_EXT.3.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator verifies that the TSS describes the method used to perform self-testing on the TSF executable code, and that this method is consistent with what is described in the SFR.

Section 6.6 of the St states the TOE ensures the integrity of its firmware image. The TOE's bootloader verifies the ECDSA signature on the main firmware image before uncompressing and executing it.

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.6.10 Trusted update (NDcPP22e:FPT_TUD_EXT.1)

2.6.10.1 NDcPP22e:FPT_TUD_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.6.10.2 NDcPP22e:FPT_TUD_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.6.10.3 NDcPP22e:FPT_TUD_EXT.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

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.

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.

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.

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.

Section 6.6 of the ST states the TOE provides an administrative command to check the installed firmware and further allows an administrator to updated the TOEs firmware by supplying a signed firmware update image. The TOE uses an existing, internal public key to verify the new image's signature (ensuring authenticity and integrity). The TOE does not support automatic checking of updates.

Component Guidance Assurance Activities: 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.

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.

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.

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.

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.

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.

In section 3.5 "Provisioning Application" of the AGD it is stated: "When the GUI is displayed, the TOE information is displayed such as the HW Version, SW Version, Serial number, and current timestamp."

The "Software Update" section of the Admin Guide explains how to install a software update. It identifies where to get the update and the process for applying the update. The software update image file needs to be a DataSoft supplied firmware image in "swupdate" format signed with DataSoft's image signing tools/keys. The TOE will automatically reboot when the software update is complete and it will be running the newly installed software.

Component Testing Assurance Activities: 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.

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.

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. 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.

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.

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).

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

Test 1 - The evaluator displayed the version of the product and then installed an update. The signature verified and the update was successfully installed.

Test 2 - The evaluator attempted to upload three images: a corrupted image, an image with an invalid signature and an image missing a signature. In each case, the TOE checks the integrity of the image after the download. The TOE correctly detects an invalid image file and rejects the download.

Test 3 - Not applicable as published hash is not used to verify the integrity of the TOE updates.

2.6.11 Trusted Update (VPNGW12:FPT_TUD_EXT.1)

2.6.11.1 VPNGW12:FPT_TUD_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.6.11.2 VPNGW12:FPT_TUD_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.6.11.3 VPNGW12:FPT_TUD_EXT.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: There is no change to the Evaluation Activities specified for this SFR in the NDcPP Supporting Document. The PP-Module modifies this SFR to mandate that a particular selection be chosen, but this selection is part of the original definition of the SFR so no new behavior is defined by the PP-Module.

See NDcPP22e:FPT_TUD_EXT.1

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.7 TOE access (FTA)

2.7.1 TSF-initiated Termination (NDcPP22e:FTA_SSL.3)

2.7.1.1 NDcPP22e:FTA_SSL.3.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

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

Section 6.7 of the ST states the TOE allows an administrator to configure a session inactivity time interval range of 3-30 minutes for inactive remote administrator sessions.

Component Guidance Assurance Activities: The evaluator shall confirm that the guidance documentation includes instructions for configuring the inactivity time period for remote administrative session termination.

Section 3.12 includes instructions for configuring the inactivity time period for remote administrative sessions.

Component Testing Assurance Activities: 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.

Test 1: The evaluator configured timeout values on the TOE. The evaluator then performed an action on the TOE followed by no further activity until after the session timeout. The evaluator observed that in each case, the session is terminated after the configured time period.

2.7.2 User-initiated Termination (NDcPP22e:FTA_SSL.4)

2.7.2.1 NDcPP22e:FTA_SSL.4.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

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

Section 6.7 of the ST state the TOE allows administrators to terminate their remote or local sessions either through the exit command or by closing their SSH session.

Component Guidance Assurance Activities: The evaluator shall confirm that the guidance documentation states how to terminate a local or remote interactive session.

Section 3.2 Account Configuration states that the command ‘exit’ can be used to log out for both local and remote sessions.

Component Testing Assurance Activities: 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.

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.

Test 1 - The evaluator established a local session and manually ‘exit’ed per the command identified in the Guidance. The session was terminated and a logout audit record was logged.

Test 2 - The evaluator established a remote session and manually ‘exit’ed per the command identified in the Guidance. The session was terminated and a logout audit record was logged. This was tested as part of NDcPP22e:FIA_UIA_EXT.1 where a logout was performed as described.

2.7.3 TSF-initiated Session Locking (NDcPP22e:FTA_SSL_EXT.1)

2.7.3.1 NDcPP22e:FTA_SSL_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.7 of the ST states the TOE does not support local administrative session locking, but does support local administrative session termination.

Component Guidance Assurance Activities: 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.

Section 3.12 “Admin Configuration” of the Admin Guide states The administrator can also set the “Session Inactivity Timeout (min)” to a value between 3 and 30 minutes (Edit Config->Admin [Tab]->Session Inactivity Timeout (min)).

The TOE does not support local administrative session locking and instead simply terminates such sessions after the administrative configured Session Inactivity Timeout.

Component Testing Assurance Activities: 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 reauthentication is needed when trying to unlock the session.

Test 1: The evaluator tested the TSF-initiated timeout of a session, by configuring the TOE timeout values for 3, 5 and 7 minutes. The evaluator then performed a login from the console and no further activity until the session was automatically closed by the TOE.

2.7.4 Default TOE Access Banners (NDcPP22e:FTA_TAB.1)

2.7.4.1 NDcPP22e:FTA_TAB.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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).

Section 6.7 of the ST states the TOE allows an administrator to set a banner that the TOE displays before each administrative session. Section 6.3 explains the administrative interfaces - The TOE provides the same SSH authentication to local and remote administrators; however, access through the TOE's Ethernet port constitutes remote administration and access through the TOE's USB interface constitutes local administration.

Component Guidance Assurance Activities: The evaluator shall check the guidance documentation to ensure that it describes how to configure the banner message.

Section 3.12 of the AGD goes over how to configure the warning banner for the TOE. Specifically it states that on the admin tab of the DPA tool a user can configure the warning banner.

Component Testing Assurance Activities: 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.

Test 1: The evaluator configured the TOE for login banners and verified with each method of access that the banner was displayed.

2.7.5 TOE Session Establishment - per TD0679 (WLANAS10:FTA_TSE.1)

2.7.5.1 WLANAS10:FTA_TSE.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that all of the attributes on which a client session can be denied are specifically defined.

Section 6.7 of the ST states the TOE allows an administrator to configure the TOE to deny wireless client sessions based upon time and day (the TOE has only a single interface).

Component Guidance Assurance Activities: The evaluator shall examine the operational guidance to determine that it contains guidance for configuring each of the attributes identified in the TSS.

The "Wi-Fi Restriction Configuration" section of the Admin Guide states the TOE can use the provided script to prevent Wi-Fi operation based on time of day or day of week

Component Testing Assurance Activities: The evaluator shall also perform the following test for each attribute:

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.

Test 1 - The evaluator configured the following: a time and day range to deny wifi access and for how long. The evaluator then attempted to establish a session in contravention to these attribute settings and in all cases observed that the access attempt failed.

2.7.6 VPN Client Management - per TD0656 (VPNGW12:FTA_VCM_EXT.1)

2.7.6.1 VPNGW12:FTA_VCM_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall check the TSS to verify that it asserts the ability of the TSF to assign a private IP address to a connected VPN client.

Section 6.7of the ST states the TOE assigns private IP addresses to connected VPN clients.

Component Guidance Assurance Activities: There are no operational guidance EAs for this component.

Component Testing Assurance Activities: The evaluator shall connect a remote VPN client to the TOE and record its IP address as well as the internal IP address of the TOE. The evaluator shall verify that the two IP addresses belong to the same network. The evaluator shall disconnect the remote VPN client and verify that the IP address of its underlying platform is no longer part of the private network identified in the previous step.

The evaluator connected a remote VPN client to the TOE and recorded its IP address with the internal IP address of the TOE. The evaluator verified that the two IP addresses belong to the same network. The evaluator disconnected the remote VPN client and verified that the IP address of its underlying platform after the disconnection was not part of the private network identified in the previous step.

2.8 Trusted path/channels (FTP)

2.8.1 Inter-TSF trusted channel - per TD0639 (NDcPP22e:FTP_ITC.1)

2.8.1.1 NDcPP22e:FTP_ITC.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.1.2 NDcPP22e:FTP_ITC.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.1.3 NDcPP22e:FTP_ITC.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.8 of the ST states the TOE secures the connections to a remote syslog server and or to an 802.1X authentication server (RADIUS server) using IKEv2/IPsec. All protocols in the SFR are addressed.

Component Guidance Assurance Activities: 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.

Section 1.5 “Operational Environment” of the Admin Guide states that Allows users to establish wireless communications with an organization’s private network through the TOE’s 802.11 Access Point and IPsec VPN.

Section 1.5 “Operational Environment” of the Admin Guide states that the purpose of the RADIUS Authentication Server is to authenticate wireless clients using EAP-TLS. FreeRADIUS 3.0.x or higher is required in the IT environment to support RADIUS communication. An IPSEC trusted channel is required to protect the RADIUS traffic.

Section 1.5 “Operational Environment” of the Admin Guide states that Any syslog server to which the TOE would transmit syslog messages over an IPSEC trusted channel.

Section 3.13 “Audit Log Configuration” of the Admin Guide provides information on how to set up syslog connection. It also states information about the behavior of the TOE when the connection is broken.

Component Testing Assurance Activities: 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.

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.

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.

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

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.

Further assurance activities are associated with the specific protocols.

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.

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.

Test 1 - The evaluator configured the TOE for external authentication using RADIUS over the tunneled IPsec Network. The evaluator also configured the TOE for syslog over the tunneled IPsec network.

Test 2 - This was tested through the FCS_IPSEC_EXT.1 tests as well as test 4 below. The TOE can be configured to connect as a client to a peer and will attempt to establish a connection.

Test 3 - This test was performed as part of test 4 below where the packet captures showed that channel data was not sent in plaintext.

Test 4 - With the connection established, the evaluator physically disconnected the network between the TOE and the remote audit/RADIUS server. After re-establishing the physical connection, the evaluator verified that the TOE reestablished the encrypted tunnel and no data was sent in plaintext. For the MAC layer timeout, the evaluator physically disrupted the connection for about 1-2 minutes. For the APP layer timeout, the evaluator physically disrupted the connection for about 5 minutes.

2.8.2 Inter-TSF Trusted Channel (WLANAS10:FTP_ITC.1)

2.8.2.1 WLANAS10:FTP_ITC.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.2.2 WLANAS10:FTP_ITC.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.2.3 WLANAS10:FTP_ITC.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator will perform the following activities in addition to those required by the NDcPP:

The evaluator will examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator will also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST.

Section 6.8 of the ST states the TOE secures the connections to a remote syslog server and or to an 802.1X authentication server (RADIUS server) using IKEv2/IPsec. All protocols in the SFR are addressed.

The TOE's typical deployment uses WPA3-SAE-PK, but the TOE also allows an administrator to configure use of WPA2 or WPA3 Enterprise. All three methods ensure a trusted channel between the TOE and its WLAN clients.

Component Guidance Assurance Activities: The evaluator will perform the following activities in addition to those required by the NDcPP:

The evaluator will 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.

Section 3.21 "Monitor and Troubleshoot" of the Admin Guide states If the TOE's Wi-Fi network is overloaded for an extended period as part of network stress testing, mission operations, etc., the periodic VPN rekey packets may timeout and the VPN client will become disconnected from the TOE, i.e., the connection will be unintentionally broken. To recover from this condition, open the VPN client on the EUD and reconnect to the TOE.

Section 3.15 "Radius Connection" of the Admin Guide states how configure Radius connection on the TOE and instructions for re-establishing a connection should a connection be unintentionally lost.

Component Testing Assurance Activities: The evaluator will perform the following activities in addition to those required by the NDcPP:

The evaluator will perform the following tests:

Test 1: The evaluator will 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.

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

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

Test 4: The evaluator will, for each protocol associated with each authorized IT entity tested during test 1, physically interrupt an established connection. The evaluator will ensure that when physical connectivity is restored, communications are appropriately protected.

Refer to NDcPP22e:FTP_ITC.1.3

2.8.3 Inter-TSF Trusted Channel (WLAN Client Communications) (WLANAS10:FTP_ITC.1/Client)

2.8.3.1 WLANAS10:FTP_ITC.1.1/Client

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.3.2 WLANAS10:FTP_ITC.1.2/Client

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.3.3 WLANAS10:FTP_ITC.1.3/Client

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: This component is adequately evaluated when performing the evaluation activities for FTP_ITC.1 in the Network Device, version 2.2e base-PP.

See NDcPP22e:FTP_ITC.1

Component Guidance Assurance Activities: This component is adequately evaluated when performing the evaluation activities for FTP_ITC.1 in the Network Device, version 2.2e base-PP.

Component Testing Assurance Activities: This component is adequately evaluated when performing the evaluation activities for FTP_ITC.1 in the Network Device, version 2.2e base-PP.

See NDcPP22e:FTP_ITC.1

2.8.4 Inter-TSF Trusted Channel (VPN Communications) (VPNGW12:FTP_ITC.1/VPN)

2.8.4.1 VPNGW12:FTP_ITC.1.1/VPN

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The EAs specified for FTP_ITC.1 in the Supporting Document for the Base-PP shall be applied for IPsec VPN communications.

Section 6.8 of the ST states the TOE uses IPsec to protect its communications with VPN clients and with the two servers described above under NDcPP22e:FTP_ITC.1 (a syslog server and an enterprise RADIUS server). The TOE acts as an IKE/IPsec responder to VPN clients and an initiator when establishing a secure connection with a syslog or RADIUS server.

Component Guidance Assurance Activities: The EAs specified for FTP_ITC.1 in the Supporting Document for the Base-PP shall be applied for IPsec VPN communications.

See NDcPP22e:FTP_ITC.1

Component Testing Assurance Activities: TheEAs specified for FTP_ITC.1 in the Supporting Document for the Base-PP shall be applied for IPsec VPN communications. Additional evaluation testing for IPsec is covered in FCS_IPSEC_EXT.1.

Test 1 - The evaluator follows the AGD and configured the TOE to accept IPsec connections from wireless (Wi-Fi) VPN clients.

Test 2 – N/A the TOE does not initiate these connections, rather the TOE waits for its VPN clients to initiate the IPsec connection

Test 3 – This evaluator captured packets transmitted between a VPN client and the TOE as part of test 4 below where the packet captures showed that channel data was not sent in plaintext.

Test 4 - With the connection established, the evaluator physically disconnected the network between the TOE and the VPN client. Because the VPN clients are also wireless (Wi-Fi) clients, the evaluator made the physical disconnect between the wireless bridge and the VPN client (connecting, through wireless bridge, to the TOE). After re-establishing the physical connection, the evaluator verified that the TOE resumed sending IPsec encrypted tunnel and no data was sent in plaintext. The evaluator repeated this test for a MAC and App layer test. For the

MAC layer timeout, the evaluator physically disrupted the connection for about 2 minutes. For the APP layer timeout, the evaluator physically disrupted the connection for about 5 minutes.

2.8.5 Trusted Path - per TD0639 (NDcPP22e:FTP_TRP.1/Admin)

2.8.5.1 NDcPP22e:FTP_TRP.1.1/Admin

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.5.2 NDcPP22e:FTP_TRP.1.2/Admin

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.5.3 NDcPP22e:FTP_TRP.1.3/Admin

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: 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.

Section 6.8 of the ST states the TOE supports remote administrators through interactive SSH sessions. This matches the selection in the SFR.

Component Guidance Assurance Activities: The evaluator shall confirm that the guidance documentation contains instructions for establishing the remote administrative sessions for each supported method.

Section 3.1 “Setup” of the Admin Guide states The TOE provides two interfaces: a data interface (Ethernet interface through pogo pins) and a local interface (acting as a USB to Ethernet peripheral). An administrator uses the SSH through local interface to provision the TOE and for local access after provisioning. An administrator can use the data interface to export audit logs post-mission (or in alternative configurations, an administrator can use the data interface to support both connections to a WPA-Enterprise/RADIUS server as well as remote SSH administrative access).

Component Testing Assurance Activities: 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.

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

Further assurance activities are associated with the specific protocols.

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.

Test 1: The evaluator configured the TOE for SSH. The evaluator connected an SSH client to the TOE and verified that the connection was successful, and that the traffic was encrypted.

Test 2: This test was performed as part of test 1 where the packet captures showed that the network traffic is protected appropriately

3. Protection Profile SAR Assurance Activities

The following sections address assurance activities specifically defined in the claimed Protection Profile that correspond with Security Assurance Requirements.

3.1 Development (ADV)

3.1.1 Basic Functional Specification (ADV_FSP.1)

Assurance Activities: The EAs for this assurance component focus on understanding the interfaces (e.g., application programing interfaces, command line interfaces, graphical user interfaces, network interfaces) described in the AGD documentation, and possibly identified in the TOE Summary Specification (TSS) in response to the SFRs. Specific evaluator actions to be performed against this documentation are identified (where relevant) for each SFR in Section 2, and in EAs for AGD, ATE and AVA SARs in other parts of Section 5.

The EAs presented in this section address the CEM work units ADV_FSP.1-1, ADV_FSP.1-2, ADV_FSP.1-3, and ADV_FSP.1-5.

The EAs are reworded for clarity and interpret the CEM work units such that they will result in more objective and repeatable actions by the evaluator. The EAs in this SD are intended to ensure the evaluators are consistently performing equivalent actions.

The documents to be examined for this assurance component in an evaluation are therefore the Security Target, AGD documentation, and any required supplementary information required by the cPP: no additional 'functional specification' documentation is necessary to satisfy the EAs. The interfaces that need to be evaluated are also identified by reference to the EAs listed for each SFR, and are expected to be identified in the context of the Security Target, AGD documentation, and any required supplementary information defined in the cPP rather than as a separate list specifically for the purposes of CC evaluation. The direct identification of documentation requirements and their assessment as part of the EAs for each SFR also means that the tracing required in ADV_FSP.1.2D (work units ADV_FSP.1-4, ADV_FSP.1-6 and ADV_FSP.1-7) is treated as implicit and no separate mapping information is required for this element.

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.

In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g. audit review or performing updates). Additionally, those interfaces that are identified in the ST, or guidance documentation, as adhering to the security policies (as presented in the SFRs), are also considered security relevant. The intent is that these interfaces will be adequately tested, and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied.

The set of TSFI that are provided as evaluation evidence are contained in the Administrative Guidance and User Guidance.

The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant.

The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

The evaluator uses the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in Section 2, including the EAs associated with testing of the interfaces.

It should be noted that there may be some SFRs that do not have an interface that is explicitly 'mapped' to invoke the desired functionality. For example, generating a random bit string, destroying a cryptographic key that is no longer needed, or the TSF failing to a secure state, are capabilities that may be specified in SFRs, but are not invoked by an interface.

However, if the evaluator is unable to perform some other required EA because there is insufficient design and interface information, then the evaluator is entitled to conclude that an adequate functional specification has not been provided, and hence that the verdict for the ADV_FSP.1 assurance component is a 'fail'.

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. No additional 'functional specification' documentation is necessary to satisfy the Evaluation Activities specified in the Supporting Document (SD).

3.2 Guidance documents (AGD)

3.2.1 Operational User Guidance (AGD_OPE.1)

Assurance Activities: 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 (per TD0536):

The evaluator performs the CEM work units associated with the AGD_OPE.1 SAR. Specific requirements and EAs on the guidance documentation are identified (where relevant) in the individual EAs for each SFR.

In addition, the evaluator performs the EAs specified below.

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 administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

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.

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.

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.

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.

b) The documentation must describe the process for verifying updates to the TOE by verifying a digital signature. The evaluator shall verify that this process includes the following steps:

1) 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).

2) 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.

Section 1.6 of the AGD identifies the evaluated configuration.

The “Software Update” section of the Admin Guide explains how to install a software update. It identifies where to get the update and the process for applying the update. The software update image file needs to be a DataSoft supplied firmware image in “swupdate” format signed with DataSoft’s image signing tools/keys. The TOE will automatically reboot when the software update is complete and it will be running the newly installed software.

3.2.2 Preparative Procedures (AGD_PRE.1)

Assurance Activities: As with the operational guidance, the developer should look to the Evaluation Activities to determine the required content with respect to preparative procedures.

It is noted that specific requirements for Preparative Procedures are defined in [SD] for distributed TOEs as part of the Evaluation Activities for FCO_CPC_EXT.1 and FTP_TRP.1(2)/Join.

The evaluator performs the CEM work units associated with the AGD_PRE.1 SAR. Specific requirements and EAs on the preparative documentation are identified (and where relevant are captured in the Guidance Documentation portions of the EAs) in the individual EAs for each SFR.

Preparative procedures are distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

In addition, the evaluator performs the EAs specified below.

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).

The documentation should be in an informal style and should be written with sufficient detail and explanation that they can be understood and used by the target audience (which will typically include IT staff who have general IT experience but not necessarily experience with the TOE product itself).

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.

The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment.

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.

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.

Section 3.0 “Preparative Procedures and Operational Guidance for the TOE” of the Admin Guide describes the TOE’s operational environment. That section and subsequent sections are written in an informal style that includes sufficient detail to address the TOE platform.

Section 3.2 “Account Configuration” of the Admin Guide describes instructions for the protected administrative capability and the default administrative account and default password.

3.3 Life-cycle support (ALC)

3.3.1 Labelling of the TOE (ALC_CMC.1)

Assurance Activities: This component is targeted at identifying the TOE such that it can be distinguished from other products or versions from the same vendor and can be easily specified when being procured by an end user. A label could consist of a 'hard label' (e.g., stamped into the metal, paper label) or a 'soft label' (e.g., electronically presented when queried).

The evaluator performs the CEM work units associated with ALC_CMC.1.

When evaluating that the TOE has been provided and is labelled with a unique reference, the evaluator performs the work units as presented in the CEM.

The evaluator verified that the ST, TOE, and Guidance are all labeled with the same hardware versions and software. The information is specific enough to procure the TOE and it includes hardware models and software versions. The evaluator checked the TOE software version and hardware identifiers during testing by examining the actual machines used for testing.

3.3.2 TOE CM Coverage (ALC_CMS.1)

Assurance Activities: Given the scope of the TOE and its associated evaluation evidence requirements, the evaluator performs the CEM work units associated with ALC_CMS.1.

When evaluating the developer's coverage of the TOE in their CM system, the evaluator performs the work units as presented in the CEM.

See section 3.3.1 above for an explanation of how all CM items are addressed.

3.4 Tests (ATE)

3.4.1 Independent Testing - Conformance (ATE_IND.1)

Assurance Activities: Testing is performed to confirm the functionality described in the TSS as well as the guidance documentation (includes 'evaluated configuration' instructions). The focus of the testing is to confirm that the requirements specified in Section 5.1.7 are being met. The Evaluation Activities in [SD] identify the specific testing activities necessary to verify compliance with the SFRs. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/TOE combinations that are claiming conformance to this cPP.

The focus of the testing is to confirm that the requirements specified in the SFRs are being met. Additionally, testing is performed to confirm the functionality described in the TSS, as well as the dependencies on the Operational guidance documentation is accurate.

The evaluator performs the CEM work units associated with the ATE_IND.1 SAR. Specific testing requirements and EAs are captured for each SFR in Sections 2, 3 and 4.

The evaluator should consult Appendix B when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

Note that additional Evaluation Activities relating to evaluator testing in the case of a distributed TOE are defined in section B.4.3.1.

The evaluator created a Detailed Test Report (DTR) to address all aspects of this requirement. The DTR discusses the test configuration, test cases, expected results, and test results. The test configuration consisted of the TOE along with supporting software and products.

Supporting Software:

- Windows 10.0
- Wireshark version 2.6.6
- Windows SSH Client – Putty version 0.76 (used to connect to device console and SSH)
- DataSoft Device Provisioning Application (DPA) version 2.0.5

The Gossamer Test servers with an Ubuntu environment acted as platforms to initiate testing. The test servers also acted as a syslog server.

- Openssh-client version 7.2p2
- Big Packet Putty version 6.2
- Nmap version 7.01
- Tcpdump version 4.9.3
- Libpcap version 1.7.4
- Stunnel version 5.30
- Openssl version 1.0.2g

- Strongswan version 5.2.5
- Rsyslog version 8.16.0

Supporting Products:

- Google Pixel 7
- Google Pixel 6
- DD-WRT (Linksys WRT54G/GL/GS)
- Kali Linux 6.0.0-kali3-amd64

3.5 Vulnerability assessment (AVA)

3.5.1 Vulnerability Survey (AVA_VAN.1)

Assurance Activities: While vulnerability analysis is inherently a subjective activity, a minimum level of analysis can be defined and some measure of objectivity and repeatability (or at least comparability) can be imposed on the vulnerability analysis process. In order to achieve such objectivity and repeatability it is important that the evaluator follows a set of well-defined activities, and documents their findings so others can follow their arguments and come to the same conclusions as the evaluator. While this does not guarantee that different evaluation facilities will identify exactly the same type of vulnerabilities or come to exactly the same conclusions, the approach defines the minimum level of analysis and the scope of that analysis, and provides Certification Bodies a measure of assurance that the minimum level of analysis is being performed by the evaluation facilities.

In order to meet these goals some refinement of the AVA_VAN.1 CEM work units is needed. The following table indicates, for each work unit in AVA_VAN.1, whether the CEM work unit is to be performed as written, or if it has been clarified by an Evaluation Activity. If clarification has been provided, a reference to this clarification is provided in the table.

Because of the level of detail required for the evaluation activities, the bulk of the instructions are contained in Appendix A, while an 'outline' of the assurance activity is provided below.

In addition to the activities specified by the CEM in accordance with Table 2, the evaluator shall perform the following activities.

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.

The developer shall provide documentation identifying the list of software and hardware components7 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 the TOE) such as a web server and protocol or cryptographic libraries. 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 hypotheses during their analysis.

If the TOE is a distributed TOE then the developer shall provide:

a) documentation describing the allocation of requirements between distributed TOE components as in [NDcPP, 3.4]

b) a mapping of the auditable events recorded by each distributed TOE component as in [NDcPP, 6.3.3]

c) additional information in the Preparative Procedures as identified in the refinement of AGD_PRE.1 in additional information in the Preparative Procedures as identified in 3.5.1.2 and 3.6.1.2.

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.

The vulnerability analysis is in the Detailed Test Report (DTR) prepared by the evaluator. The vulnerability analysis includes a public search for vulnerabilities and fuzz testing. None of the public search for vulnerabilities, or the fuzz testing uncovered any residual vulnerability.

The evaluation team performed a public search for vulnerabilities in order to ensure there are no publicly known and exploitable vulnerabilities in the TOE from the following sources:

- National Vulnerability Database (https://web.nvd.nist.gov/vuln/search)
- Vulnerability Notes Database (http://www.kb.cert.org/vuls/)
- Rapid7 Vulnerability Database (https://www.rapid7.com/db/vulnerabilities)
- Tipping Point Zero Day Initiative (http://www.zerodayinitiative.com/advisories)
- Exploit / Vulnerability Search Engine (http://www.exploitsearch.net)
- SecurITeam Exploit Search (http://www.securiteam.com)
- Tenable Network Security (http://nessus.org/plugins/index.php?view=search)
- Offensive Security Exploit Database (https://www.exploit-db.com/)

The search was performed on 7/19/2023 with the following search terms:

- "RAP 117"
- Datasoft
- openssl 3.1.0

- Radio Access Point
- strongswan
- rsyslog
- Freescale
- Freescale i.MX6UL rev1.2 696 MHz
- Linux kernel 5.4
- wpa2
- wpa3
- TCP
- IPSEC
- "sidebridge"

3.5.2 Additional Flaw Hypotheses (AVA_VLA.1)

Assurance Activities: The following additional tests shall be performed:1.) [Conditional]: If the TOE is a TLS server and supports ciphersuites that use RSA transport (e.g. supporting TLS_RSA_WITH_* ciphers) the following test shall be performed. Where RSA Key Establishment schemes are claimed and especially when PKCS#1 v1.5* padding is used, the evaluators shall test for implementation flaws allowing Bleichenbacher and Klima et al. style attacks, including Bock et al's ROBOT attacks of 2017 in the flaw analysis. Even though Bleichenbacher's original paper is two decades old, Bock et al. found these attacks to still be effective in weakening the security of RSA key establishment in current products. Bleichenbacher and Klima et al. style attacks are complex and may be difficult to detect, but a number of software testing tools have been created to assist in that process. The iTC strongly recommends that at least one of the tools mentioned in Bock et al's ROBOT attacks of 2017 webpage or paper, as effective to detect padding oracle attacks, be used to test TOE communications channels using RSA based Key Establishment (related sources: http://archiv.infsec.ethz.ch/education/fs08/secsem/bleichenbacher98.pdf, https://eprint.iacr.org/2003/052, https://robotattack.org/). Network Device Equivalency Considerations.

This test is not applicable. The TOE is not a TLS server.