Assurance Activities Report
For a Target of Evaluation

A10 Networks Thunder Series Appliances TH-4435, TH-5840-11, TH-7445, TH-7650-11, TH-7655 with ACOS 5.2.1-P3

TOE Hardware Models: TH-4435, TH-5840-11, TH-7445, TH-7650-11, TH-7655
TOE Software Version: ACOS 5.2.1-P3
TOE OS Kernel Version: Linux 4.19 LTS

collaborative Protection Profile for Network Devices Version 2.2e
Date: 2020.03.27

A10 Networks Thunder Series Appliances TH-4435, TH-5840-11, TH-7445, TH-7650-11, TH-7655 with ACOS 5.2.1-P3 AAR, v.1.0

January 25, 2023

Evaluated by:

Advanced Data Security, LLC

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

The Developer of the TOE:

A10 Networks, Inc.
2300 Orchard Parkway
San Jose, CA 95131

The TOE Evaluation was sponsored by:

A10 Networks, Inc.
2300 Orchard Parkway
San Jose, CA 95131

Evaluation Personnel:

Eugene Polulyakh
Diana Polulyakh
Valeriy Polulyakh

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

Common Evaluation Methodology Version

Common Methodology for Information Technology Security Evaluation,
Evaluation Methodology,
CCMB-2017-04-004, Version 3.1, Revision 5, April 2017

TABLE OF CONTENTS

TECHNICAL DECISIONS

TD #	TITLE	APPLIC.	EXCLUSION RATIONAL (IF APPLIC)
0670	NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	No	The TOE does not claim TLSC or DTLSC
0639	NIT Technical Decision for Clarification for NTP MAC Keys	No	The TOE uses IPsec to protect NTP traffic.
0638	NIT Technical Decision for Key Pair Generation for Authentication	No	The TOE is not a Distributed TOE.
0636	NIT Technical Decision for Clarification of Public Key User Authentication for SSH	No	The TOE does not claim SSHC
0635	NIT Technical Decision for TLS Server and Key Agreement Parameters	No	The TOE does not claim TLSS
0634	NIT Technical Decision for Clarification required for testing IPv6	No	The TOE does not claim TLSC or DTLSC
0633	NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	Yes	
0632	NIT Technical Decision for Consistency with Time Data for vNDs	No	The TOE is not a vND.
0631	NIT Technical Decision for Clarification of public key authentication for SSH Server	No	The TOE does not claim SSHS
0592	NIT Technical Decision for Local Storage of Audit Records	No	The TOE does not FAU_STG.1, FAU_STG_EXT.2/LocSpace, or FAU_STG_EXT.3/LocSpace.
0591	NIT Technical Decision for Virtual TOEs and hypervisors	No	The TOE is not virtual and does not include a hypervisor
0581	NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	No	The TOE does not claim elliptic curve-based key establishment.
0580	NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
0572	NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	No	The TOE does not claim TLSC or DTLSC
0571	NiT Technical Decision for Guidance on how to handle FIA_AFL.1	No	TOE devices all distinguish physical consoles
0570	NiT Technical Decision for Clarification about FIA_AFL.1	Yes	
0569	NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	No	The TOE does not claim TLSS or DTLSS
0564	NiT Technical Decision for Vulnerability Analysis Search Criteria	Yes	
0563	NiT Technical Decision for Clarification of audit date information	Yes	

0556	NIT Technical Decision for RFC 5077 question	No	The TOE does not claim TLSS
0555	NIT Technical Decision for RFC Reference incorrect in TLSS Test	No	The TOE does not claim TLSS
0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	
0546	NIT Technical Decision for DTLS - clarification of Application Note 63	No	The TOE does not claim DTLS
0538	NIT Technical Decision for Outdated link to allowed-with list	No	No packages claimed
0537	NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	No	The TOE does not claim DTLS
0536	NIT Technical Decision for Update Verification Inconsistency	Yes	
0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	Yes	
0527	Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes	

1. TOE OVERVIEW

The Target of Evaluation (TOE) is A10 Networks, Inc. Thunder-series of network appliances executing A10's Advanced Core Operating System (ACOS) 5.2.1-P3, a network device as defined in NDcPPv2.2e. A10 Thunder-series appliances are devices to provide organizations with networking services and capabilities including:

- Application Delivery Control (ADC),
- Carrier-Grade Networking (CGN),
- Convergent Firewall (CFW), and
- SSL Insight (SSLi)

Thunder ADC features provide a performant solution that enables customer enterprise and web applications to be highly available, accelerated and secure.

Thunder CGN features provide a performant and transparent network address and protocol translation that allows service providers and enterprises to extend IPv4 network connectivity while simultaneously transitioning to IPv6 standards.

Thunder CFW incorporates multiple security functions for enterprise and service provider deployments, including scalable and performant firewall, IPsec VPN, secure web gateway, Carrier-Grade (CG) Networks Address Translation (NAT) with integrated DDoS protection and traffic steering.

Thunder SSLi is a comprehensive SSL/TLS decryption solution that enables security deployments to efficiently analyse all enterprise traffic, ensuring compliance and privacy, and increasing performance of the organization's security stack.

Thunder ACOS supports a 64-bit, multi-CPU architecture built from the ground up to provide ADC, CGN, CFW, and SSLi services with high performance, scalability and reliability.

The following Figure 1 depicts the TOE boundary. As shown, an A10 Thunder-series appliance is a single hardware device that has management ports and network (or data plane) ports.

The TOE interfaces with the following non-TOE systems in its operational environment.

- local and remote administrative interfaces,
- Syslog server interface for external audit log storage,
- Network Time Protocol (NTP) server interface for reliable time information in audit records, and
- File server interface for trusted updates and configuration backups

The TOE also interfaces with a Certification Authority (CA) for server certificates and certificate validation using Certificate Revocation Lists (CRLs) and the Online Certificate Status Protocol (OCSP).

The TOE was evaluated as a standalone network device only. ACOS ADC, CGN, CFW, and SSLi data plane functions, while included in the product, were not evaluated during the TOE's evaluation under the NDcPP.

Only the functionality described in Section 5 of this Security Target is considered to be within the logical boundary of the TOE.

Architectural Description of the TOE

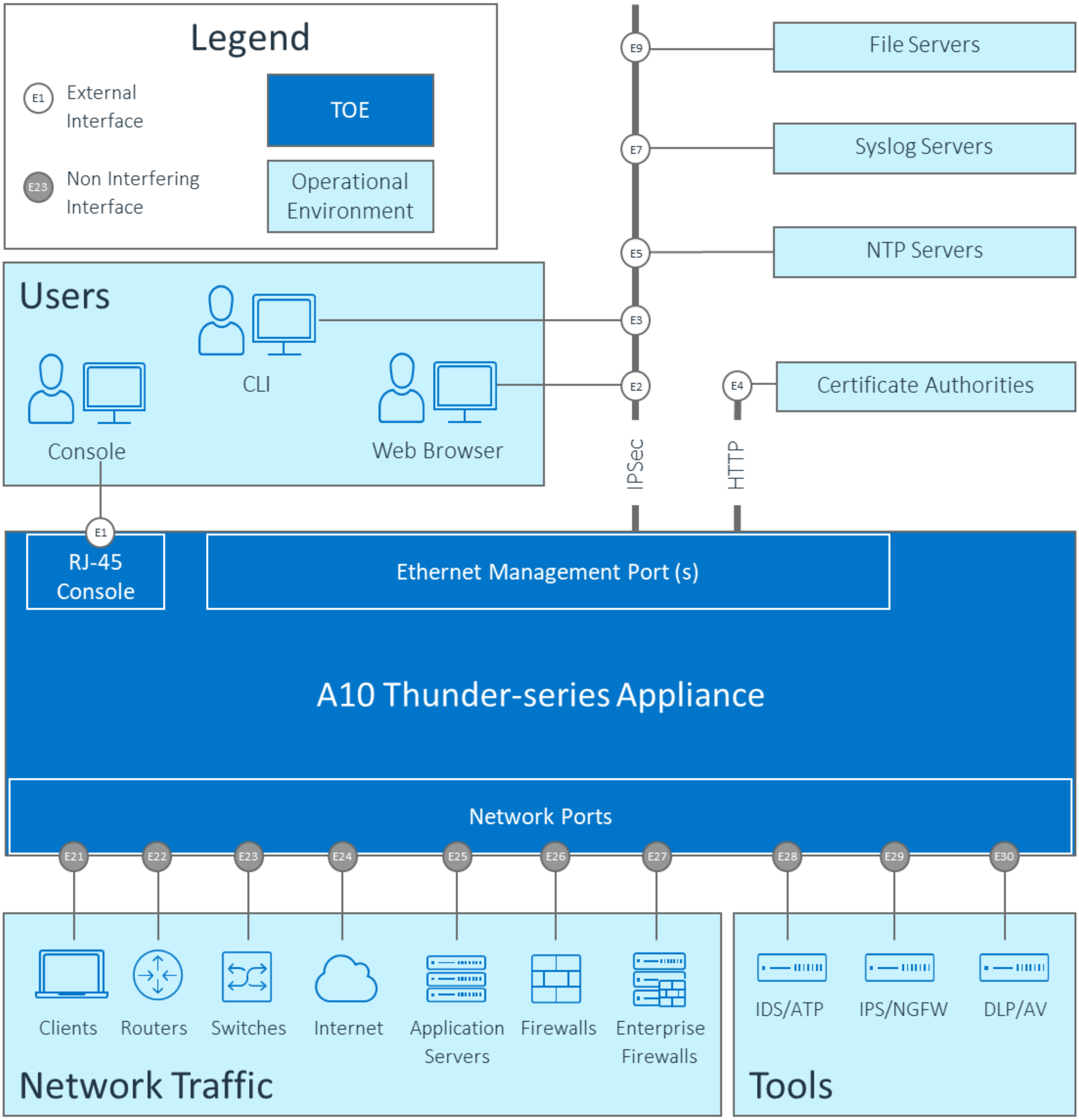


Figure 1: TOE Boundary for A10 Thunder-series Appliance

The TOE is a standalone network device. The hardware and firmware components of the TOE are enclosed in a metal enclosure which is the physical boundary of the TOE.

The scope of each TOE appliance begins with a hardware appliance having physical connections to the deployed network environment. Within the appliance, ACOS is designed to control and enable access to the available functions (e.g., program execution, device access, facilitate device functions and capabilities). ACOS enforces applicable capability and security policies on network information flowing through the appliance.

By their nature TOE appliances are administratively closed systems, providing access only through ACOS defined interfaces (e.g. CLI and Web GUI) to administrators configured in ACOS for this purpose. TOE appliances do not expose OS Shell access to administrators.

At system start-up the system control is transferred from flash memory to dynamic memory under the control of ACOS using a built-in bootstrap. ACOS reads the configuration parameters from the configuration file in non-volatile memory and then initializes the necessary data structures in dynamic memory and begins operation.

During normal operation, IP packets are sent to the management IP address or through the appliance over one or more of its physical network interfaces. The appliance processes them according to the system's configuration and state information dynamically maintained by the appliance. This processing typically results in the data plane packets being forwarded out of the device over another interface. The TOE will process control and management plane packets destined for the TOE based on the requirements of the given protocol (e.g. IPSec, OSPF, etc.).

List of Acronyms

ACRONYM/ABBREVIATION	MEANING
ACL	Access Control List
ACOS	Advanced Core Operating System
ADC	Application Delivery Controller
AES	Advanced Encryption Standard
AH	Authentication Header
ATP	Advanced Threat Protection
AV	Anti-Virus
CA	Certificate Authority
CBC	Cipher Block Chaining
CC	Common Criteria
CFW	Convergent Firewall
CGN	Carrier-Grade NAT
CLI	Command-line interface
CMVP	Cryptographic Module Validation Program
CRL	Certificate Revocation List
CSP	Critical Security Parameter
CSR	Certificate Signing Requests
DDoS	Distributed Denial of Service
DH	Diffie-Hellman
DLP	Data Loss Prevention
DNS	Domain Name System
ECDSA	Elliptic Curve Digital Signature Algorithm
ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standards
GCM	Galois Counter Mode
GUI	Graphical User Interface
HMAC	Keyed-Hash Message Authentication Code
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPsec	Internet Protocol Security
IPS	Intrusion Protection System
IT	Information Technology

ACRONYM/ABBREVIATION	MEANING
NAT	Network Address Translation
NDcPP	Network Device collaborative Protection Profile
NGFW	Next Generation Firewall
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
OS	Operating System
OCSP	Online Certificate Status Protocol
PP	Protection Profile
PKI	Public Key Infrastructure
PRF	Pseudo Random Function
PSK	Pre-Shared Key
PUBS	Publications
QSFP	Quad (4-channel) Small Form-factor Pluggable
QSFP28	Quad (4-channel) Small Form-factor Pluggable 28 GB data
RADIUS	Remote Authentication Dial-In User Service
RBG	Random Bit Generator
RSA	Rivest Shamir Adleman Algorithm
SA	Security Associations (IPSec)
SCP	Secure Copy Protocol
SD	Supporting Document
SECP	Standards for Efficient Cryptography Parameter
SF	Security Function
SFP	Security Function Policy
SFP	Small Form-factor Pluggable
SFTP	Secure File Transfer Protocol
SHA	Secure Hash Algorithm
SSD	Solid State Disk
SSH	Secure Shell
SSLi	SSL Intercept
ST	Security Target
TACACS	Terminal Access Controller Access-Control System
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSS	TOE Summary Specification
VPN	Virtual Private Network

2. TOE SUMMARY SPECIFICATION ASSURANCE ACTIVITIES

2.1 FAU: Security audit

FAU_GEN.1

TSS

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.

Verdict: Pass

Section 7.1.1.1 states that the TOE generates audit records for events including:

- starting and stopping the audit function,
- administrator commands, and
- all other events identified in Section 5.2.1.1.

Audit records include the date/time, event source (CLI/GUI), responsible administrator (user), IP address of the administrator, and additional event-specific content indicated in Section 5.2.1.1. CLI management operations are distinguished between local console and remote management in logged records by the indicated IP address, with:

• Local Console Operations	Loopback IP Address	(e.g. 127.0.0.1)
• Remote Management Operations	Peer IP Address and TCP Port	(e.g. 10.65.25.166:53288)

The successful outcome of events is implicit, without the adjacent event records indicating failure.

Importing, deleting, and generating cryptographic keys is audited including corresponding key name identifiers to identify the keys involved for such operation.

FAU_GEN.2

TSS

(This is already covered by the TSS requirements for FAU_GEN.1.)

Verdict: Pass

FAU_STG_EXT.1

TSS

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

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.

Verdict: Pass

Section 7.1.1.3 states that the generated audit records are stored in two files on the local filesystem of the standalone TOE, one for ACOS audit category records and one for ACOS event category records. Each record file supports a circular logging store with the oldest records overwritten first when the logging store is full.

The audit category logging store is configurable from 1000 to 30,000 records with 20,000 records as a default. The event category record logging store is configurable from 10,000 to 50,000 records with 30,000 records as a default. Only authorized administrators can enable/disable logging to the logging stores, clear the content of the logging stores, or alter the sizes of the logging stores. All TOE administrators are permitted to view contents of the logging stores. No other methods are supported by the TOE to change the content of records in the TOE's local audit stores.

The TOE can be configured to simultaneously report audit records for both logging stores to external SYSLOG servers in real-time, protected through the use of IPsec. Only authorized administrators can add, delete, or modify SYSLOG configuration settings.

2.2 FCS: Cryptographic support

FCS_CKM.1

TSS

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.

Verdict: Pass

Section 7.1.2.1 states that in support of secure cryptographic protocols, the TOE supports RSA key generation schemes to be used with IPsec, in accordance with PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 using 2048 bit keys.

The TOE also supports Elliptic Curve key generation schemes using the P-256, P-384 curves to be used in IPsec certificates in accordance with FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4.

The TOE implements FFC schemes using Diffie-Hellman group 14 for IPsec. The TOE implementation of Diffie-Hellman group 14 (2048 MODP) meets RFC 3526, Section 3.

The TOE also key generation using RSA for 2048-bit keys and ECDSA with P-256/P-384 curves when creating keypairs as part of Certificate Signing Request (CSR) generation.

The relevant key generation algorithms are validated under CAVP certificates # C1198 and C1940.

Verdict: Pass

FCS_CKM.2

TSS ***[TD 0580]***

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.

Verdict: Pass

See discussion in FCS_CKM.1 above.

FCS_CKM.4

TSS

The evaluator examines the TSS to ensure it lists all relevant keys (describing the origin and storage location of each), all relevant key destruction situations (e.g. factory reset or device wipe function, disconnection of trusted channels, key change as part of a secure channel protocol), and the destruction method used in each case. For the purpose of this Evaluation Activity the relevant keys are those keys that are relied upon to support any of the SFRs in the Security Target. The evaluator confirms that the description of keys and storage locations is consistent with the functions carried out by the TOE (e.g. that all keys for the TOE-specific secure channels and protocols, or that support FPT_APW.EXT.1 and FPT_SKP_EXT.1, are accounted for) [Where keys are stored encrypted or wrapped under another key then this may need to be explained in order to allow the evaluator to confirm the consistency of the description of keys with the TOE functions]. 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.

Verdict: Pass

The statement in Section 7.1.2.3 points to Table 12.

FCS_COP.1/DataEncryption

TSS

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.

Verdict: Pass

Section 7.1.2.4 states that the TOE provides encryption and decryption capabilities using 128, 192, and 256 bit AES in both CBC and GCM modes. AES is implemented in the following protocols: IPsec.

These AES algorithm meet ISO/IEC 10118-3:2004 with AES-CBC meeting ISO 10116, and AES-GCM meeting ISO 19772. These algorithms are validated under CAVP certificates # C1198, C1940, and A1181.

FCS_COP.1/SigGen

TSS

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

Verdict: Pass

Section 7.1.2.5 states that the IPsec tunnels can be configured to use RSA key sizes 2048 bits (or greater) or ECDSA with key size 256 and 384 bits using NIST curve P256 and P384 certificate for IPsec authentication. For verification of trusted updates, digital signatures use an RSA key size of 2048 bits.

FCS_COP.1/Hash

TSS

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.

Verdict: Pass

Section 7.1.2.6 states that the TOE provides cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512 with message digest sizes of 160, 256, 384, and 512 bits respectively. The TSF uses hashing services the following functions:

- SHA-1, SHA-256, SHA-384, and SHA-512 for IPsec data integrity and authentication, and digital signatures.
    - Relevant algorithms include: HMAC, RSA, and ECDSA

- SHA-256 for trusted update
    - Relevant algorithms include: RSA

- SHA-256 for password hashing

The SHA algorithm meets ISO/IEC 10118-3:2004 and is validated under CAVP SHS certificates # C1198, C1940, and A1181.

In addition, the TOE use MD5 hashing services for firmware integrity checking at device boot-time.

FCS_COP.1/KeyedHash

TSS

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.

Verdict: Pass

Section 7.1.2.7 states that the TOE provides keyed-hashing message authentication services in IPsec using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with key sizes and 160, 256, 384, and 512 bits and digest sizes of 160, 256, 384, and 512 bits as specified in FIPS PUB 198-1 and FIPS PUB 180-4.

The block size is 512 bits for HMAC-SHA-1 and HMAC-SHA-256 algorithms and 1024 bits for HMAC-SHA-384 and HMAC-SHA-512 algorithms. The algorithm meets ISO/IEC 9797-2:2011 Section 7 and is validated under CAVP HMAC certificates #C1198, C1940, and A1181.

FCS_RBG_EXT.1

TSS

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.

Verdict: Pass

Section 7.1.2.8 states that the TOE implements a NIST-approved deterministic random bit generator (DRBG). The DRBG used by the TOE is CTR_DRBG with derivation function (DF) with AES. The TOE’s DRBG implementation meets ISO/IEC 18031:2011 and is validated under CAVP certificates #C1198 and C1940.

The DRBG is seeded from the hardware entropy source (Intel RNG) of the underlying TOE CPU(s). Entropy from the noise source is extracted, conditioned and used to seed the DRBG with 256 bits of full entropy.

2.3 FIA: Identification and authentication

FIA_AFL.1

TSS

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).

Verdict: Pass

Section 7.1.3.1 states that the TOE provides a counter that is incremented for consecutive failed remote authentication attempts via CLI or Web/GUI and will lock an administrator account when the failure counter threshold is reached. A valid login that occurs prior to the failure counter reaching its threshold will reset the counter to zero.

When this threshold is reached for a given administrator account, no authentication will be allowed for the account as the TOE's remote CLI and Web/GUI interfaces will refuse connections for the account from any endpoint in subsequent attempts. Once a connection is refused due to lockout for a given account, the administrator would have to re-login after the configurable lockout time period has elapsed or after a manual unlocking of the account is performed.

This lockout threshold of the TOE can be configured to any value from 1 - 10 failures. The lockout duration is a configurable number of minutes from 0 - 1440 minutes (24 hours), with a value of 0 to requiring manual unlocking by the root (master) administrator of the TOE via the local console.

Authentication failure counting is disabled for access attempts on to the local console of the TOE to ensure that remote administrator authentication failures do not lead to a situation where no administrator access is available.

FIA_PMG_EXT.1

TSS

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 charters supported for administrator passwords.

Verdict: Pass

Section 7.1.3.2 states that passwords maintained by the TOE can be composed using any combination of upper-case and lower-case letters, numbers, and special characters including the following. These password special characters are supported both for CLI and Web/GUI interfaces to successfully authenticate with the TOE.

- “ ”, “!”, “@”, “#”,“$”, “%”, “^”, “&”, “*”, “(”, “)”, “ ”, “"”, “'”, “+”, “,”, “-”, “.”, “/”, “:”, “;”, “<”, “=”, “>”, “?”, “[”, “\”, “]”, “_”, “`”, “{”, “|”, “}”, “~”

The password policy is configurable by TOE administrators and supports the minimum password length of 8 characters and a maximum password length of 63 characters.

FIA_UIA_EXT.1

TSS

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.

Verdict: Pass

Section 7.1.3.3 states that the TOE requires all administrators to be successfully identified and authenticated before any TSF-mediated actions are allowed to be performed, with the only exception being the display of a banner. A successful authentication is determined by a successful username and password combination accepted by the TOE. The configurable warning banner is displayed prior to the user being prompted to enter the username component of the credential when logging in to the TOE. Accordingly, the TOE does not allow a user to perform any other actions prior to authentication.

The administrator logs into the TOE through either the local TOE console using the CLI or remotely through IPsec using the CLI via SSH or the Web/GUI via HTTPS over IPsec.

FIA_UAU_EXT.2

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.

FIA_UAU.7

TSS

None.

2.4 FMT: Security management

FMT_MOF.1(1)/ManualUpdate

TSS

There are no specific requirements for non-distributed TOEs.

FMT_MTD.1/CoreData

TSS

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.

Verdict: Pass

Section 7.1.4.2 states that security management of the TOE is restricted to authorized administrators (users). The TOE does not support non-administrative users. All users of the TOE are considered administrative users. The only function of the TOE prior to a successful authentication (username and password combination) is the display of a previously configured banner. Administrative functions of the TOE are available only after successful authentication (login).

As described in 7.1.3. SF.Identification_Authentication above, only read+write privileged administrators can create, modify, or delete TOE configuration elements. This includes the generating and exporting of X.509 Certificate Signing Requests (CSRs) as well as importing and deleting of X.509 certificates. The TOE implementation does not support modifying (changing) X.509 certificates extant on the TOE.

FMT_SMF.1

TSS (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.

Verdict: Pass

Section 7.1.4.3 states that the Administrators can log into the TOE through either the local TOE console using the CLI or remotely through IPsec using the CLI via SSH or the Web/GUI via HTTPS.

The ability to perform security management functions on the TOE is restricted to authorized administrators as indicated above. All security management functions are available both locally and remotely to authorized, authenticated administrators.

FMT_SMR.2

TSS

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.

Verdict: Pass

Section 7.1.4.4 states that the TOE provides administrative access to perform security management functions via

- local, serial console CLI
- remote, CLI
- remote, Web/GUI

The TOE maintains administrative user roles. Authorized administrators authenticated by TOE with the “read+write” privilege can perform security management functions not otherwise restricted to the root (master) administrator. Administrators with “read-only” privilege may view security management configuration settings and information, however they cannot create, import, add, or otherwise modify configuration settings.

2.5 FPT: Protection of the TSF

FPT_SKP_EXT.1

TSS

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.

Verdict: Pass

Section 7.1.5.1 states that the TOE is designed specifically to prevent access to locally-stored cryptographically protected administrative passwords or any keys stored in the TOE. The TOE does not implement any functions that will disclose to any administrator a stored cryptographic key or password. By its nature the TOE is an administratively closed systems, providing access only through defined interfaces (e.g. CLI, Web/GUI) to administrators configured to permit management access to the TOE. The TOE does not expose OS Shell access to administrators. See Table 12 for more information about keys stored on the TOE.

The TOE protects user administrative passwords by saving a SHA-256 hash of the password.

FPT_APW_EXT.1

TSS

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.

Verdict: Pass

See TOE summary specifications for FPT_SKP_EXT.1 above.

FPT_TST_EXT.1

TSS

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.

Verdict: Pass

Section 7.1.5.3 states that the TOE includes a suite of self-tests that execute at power-on or a reboot of the TOE to ensure the proper functioning of the TOE. These tests ensure that the integrity of the TOE firmware is maintained and verify cryptographic algorithms on the underlying processor match known, expected results. If any of the tests fail, the TOE will enter a limited operations mode until an Administrator intervenes. In this mode, the TOE will provide a nominal level of services sufficient to support local console CLI management access only for assessment and remediation of the failure.

Algorithm tests and firmware integrity tests include:

- AES Known Answer Test using GCM and ECB modes (encrypt/decrypt)
- SHS Known Answer Test as a part of the HMAC KAT. Also SHA-1 and SHA-256 are tested separately.
- HMAC Known Answer Test using SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 to also cover SHA POST
- SP800-90A DRBG Known Answer Test for CTR_DRBG: AES
- RSA Known Answer Test using 2048 bit key, SHA-256
- ECDSA Pairwise Consistency Test (sign/verify) using P-224, K-233 and SHA-512
- Firmware Integrity Test using MD5 hash of the firmware image

The firmware integrity test is sufficient to ensure that the TOE has not been corrupted and the algorithm tests are sufficient to ensure that the cryptographic functions are operating properly.

FPT_TUD_EXT.1

TSS

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.

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.

Verdict: Pass

Section 7.1.5.4 states that the TOE provides functions to query the version of firmware on the TOE and to upgrade the firmware embedded in the TOE device. When installing updated firmware, RSA digital signatures and image integrity checks are used to validate the update and ensure it is an update intended and originated by A10 Networks.

The TOE firmware version can be queried by a 'show version' CLI command or by viewing the TOE's device information page in the Web/GUI. Firmware updates that succeed become active upon a reboot of the TOE device.

TOE update images are available from the A10 Networks support web site (https://support.a10networks.com). An A10 Networks support login ID and password is required to download these images. To perform an update an administrator will download a TOE update image to a local, trusted file server accessible to the TOE via IPsec. The administrator will then attempt to manually update using the 'upgrade' CLI command or equivalent Web/GUI operation.

This command (operation) will prompt the TOE to download the update image to the TOE device. The TOE will then verify the image integrity and RSA digital signature of the upgrade image before installation. If the image integrity and signature verifications succeed, the TOE will proceed with installation of the firmware update. Otherwise, the TOE will cease the update operation after deleting the temporary image and logging the trusted update failure.

FPT_STM_EXT.1

TSS

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.

Verdict: Pass

Section 7.1.5.5 states that the TOE provides a source of date and time information used in audit event timestamps. The clock function is reliant on the system clock provided by the underlying hardware. The TOE can optionally be set to receive clock updates from one or more NTP servers to synchronize the TOE with the network date and time.

This date and time of the TOE is used for timestamps applied in TOE generated audit records and is also used to track inactivity of administrative sessions, track administrator account lockout times, and support cryptographic operations based on time/date.

The TOE allows authorized administrators to set the date and time manually and/or configure NTP time sources for the TOE.

2.6 FTA: TOE Access

FTA_SSL_EXT.1

TSS

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.

Verdict: Pass

Section 7.1.6.1 states that the TOE terminates sessions to the local console that have been inactive for an administrator-configured period of time. Administrators can use the ‘terminal idle-timeout’ CLI command or equivalent Web/GUI operation to configure the inactivity period after which a session to the local console will be terminated by the TOE. The inactivity period is a configurable number of minutes from 1 - 60 minutes (1 hour).

FTA_SSL.3

TSS

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

Verdict: Pass

Section 7.1.6.2 states that the TOE terminates remote, administrative sessions that have been inactive for an administrator-configured period of time. The same ‘terminal idle-timeout’ CLI command or equivalent Web/GUI operation configures the inactivity period after which an inactive remote session to the TOE CLI using SSHv2 via IPsec will be terminated by the TOE. The inactivity period is a configurable number of minutes from 1 - 60 minutes (1 hour).

Administrators can use the ‘web-service gui-timeout-policy idle’ CLI command or equivalent Web/GUI operation to configure the inactivity period after which an inactive remote Web/GUI session using HTTPS via IPsec will be terminated. The inactivity period is a configurable number of minutes from 1 - 60 minutes (1 hour).

FTA_SSL.4

TSS

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

Verdict: Pass

Section 7.1.6.3 states that the TOE allows interactive sessions to exit (logout) gracefully by command (operation). Administrative sessions to the TOE CLI using the local console or remotely accessing the CLI with SSHv2 via IPsec can terminate their sessions using the ‘exit’ CLI command.

Administrative sessions to the TOE Web/GUI via IPsec can terminate their sessions using the provided ‘logout’ option in the user control menu at the top right corner of the browser window.

FTA_TAB.1

TSS

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).

Verdict: Pass

Section 7.1.6.4 states that the TOE can be configured for advisory and consent banners displayed to interactive administrators accessing the TOE such that administrators may terminate these session before performing any function or operation on the TOE.

For local and remote CLI sessions to the TOE, a banner to be displayed before entering login credentials (username and password) can be configured using the ‘banner login’ CLI configuration command or equivalent Web/GUI operation. This banner is displayed before prompting for input credentials by the administrator.

For remote Web/GUI sessions to the TOE, a banner to be displayed before entering login credentials (username and password) can be configured on the ‘System -> Settings -> Web’ Web/GUI page. This banner is displayed in a pop-up window that must be accepted by clicking an “OK” button before the TOE’s standard login page is displayed prompting inputs for username and password.

Section 7.1.4.4 states that the TOE provides administrative access to perform security management functions via

- local, serial console CLI
- remote, CLI
- remote , Web/GUI

2.7 FTP: Trusted path/channels

FTP_ITC.1

TSS

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.

Verdict: Pass

Section 7.1.7.1 states that the TOE must be configured for IPsec tunnel(s) to support the trusted channel communications with the entities listed above secured by and tunneled within IPsec sessions established using X.509 certificates or pre-shared keys. Trusted channels initiated by the TOE are to:

- SYSLOG Servers
- File Servers
- NTP Servers

FTP_TRP.1/Admin

TSS

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.

Verdict: Pass

Section 7.1.7.2 states that the TOE supports IPsec to ensure secured communications with by remote administrators. Such remote sessions are protected from disclosure or modifications of exchanged data in the performance of administration actions and require administrator authentication before performing any functions on the TOE.

Administrative clients use the TOE CLI with SSHv2 tunneled over IPsec or the Web/GUI with HTTPS tunneled over IPsec.

2.8 FCS: Cryptographic support (selection-based)

FCS_IPSEC_EXT.1

TSS

FCS_IPSEC_EXT.1.1

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.

Verdict: Pass

Section 7.1.2.9 states that the TOE supports IPsec tunnel-mode (conformant to RFCs 4301) for trusted channel communications with SCP/SFTP Servers (for file transfers to/from the TOE and updates of the TOE), NTP Servers (for network date and time synchronization) and SYSLOG Servers (for audit logging). Trusted path communications are also supported using IPsec tunnel-mode for remote administration of the TOE by CLI (SSH) management clients and Web/GUI (HTTPS) management browser clients.

The TOE supports for ESP protection with AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128, AES-GCM-192 and AES-GCM-256 using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512.

The TOE implements IKEv2 with:

- IKEv2 as defined in RFCs 5996 (including support for NAT traversal) and RFC 4868 for hash functions.

The TOE supports IKEv2 protection with AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128, AES-GCM-192, and AES-GCM-256.

The TOE supports configuration of IKE lifetimes as follows. The TOE will initiate SA renegotiation prior to expiration or consumption of these lifetimes.

- IKEv2 SA lifetime by length of time (300 seconds to 24-hours (86400 seconds))
- IKEv2 Child SA lifetimes of number of bytes (1 or unlimited MBytes)
- IKEv2 Child SA lifetimes length of time (300 seconds to 8-hours (28800 seconds)).
-

The TOE supports Diffie-Hellman Group 14 only.

The TOE uses AES-CTR DRBG to generate the secret value ‘x’ used in the IKEv2 Diffie-Hellman key exchange (“x” in g^x mod p). These exponents generated for DH Group 14 are 224-bits (28 bytes) long. Nonces generated using AES-CTR DRBG are 256-bits (32-bytes), which is more than 128-bits in size and half of the largest output size supported (SHA-512).

The TOE ensures that the strength of the symmetric algorithm negotiated to protect IKEv2 IKE SA connections is greater than or equal to the strength of the symmetric algorithm negotiated to protect the IKEv2 Child SA connections. During establishment of IKEv2 Child SA connection, the negotiated strength of IKEv2 IKE SA is compared against that configured in the TOE for IKEv2 Child SA connection. If the configured strength is greater than that negotiated strength, than the IPsec tunnel establishment is failed and a corresponding event is logged.

The TOE supports peer authentication with RSA and ECDSA for use with X.509v3 certificates (conformant to RFC 4945) or pre-shared keys for IPsec IKE. Pre-shared keys are manually entered as text-based strings when configuring IPsec tunnels via the CLI or Web/GUI administration interfaces with a length of 1 – 127 characters. Pre-shared keys entered are combinations of lower and upper-case characters, numeric digits, and supported special characters. For a given IPsec peer’s IKE configuration using a pre-shared key, the key must be the same on the configured IPsec peer as entered by the TOE administrator.

The TOE will attempt match an FQDN or IP Address value configured by an authorized administrator for the reference identifier of a given IPsec peer with values of the Subject or Subject Alternative Name (SAN, if present) fields in the peer’s certificate when establishing an IKE session with the peer. If neither of the fields match the configured reference identifier, the IPsec tunnel establishment is failed, and a corresponding event is logged.

The TOE’s IPsec Security Policy Database (SPD) is configured based upon the trusted paths and trusted channels protected using IPsec. IPsec SPD rules are configured using firewall-style Access Control Lists (ACLs) based IP addresses/ranges, L4 protocol, L4 port/selector, and precedence. When applied IPsec endpoints or endpoints via IPsec gateways/VPNs, ACL deny and permit rules effect SPD DROP and PROTECT actions for corresponding network traffic; respectively. Permit ACLs for other endpoints effect SPD BYPASS actions as they will allow network traffic to (from) the TOE unsecured by IPsec on the TOE.

TOE ACLs are processed with precedence of the order they are specified with by TOE administrators. A default discard rule must be defined by the TOE administrators to ensure that a packet not matching a prior ACL rule is discarded (dropped). TOE administrators are also responsible for ensuring that TOE ACLs do not overlap or conflict.

Incoming IPsec traffic received on the TOE management interface is decrypted and filtered through the configured ACLs with packets allowed for further processing by the TOE software elements. Outgoing traffic on the TOE management interface is first filtered through the configured ACLs for permitted matches with IPsec peers which are then allowed for IPsec processing or direct transmission. Such outgoing traffic then matched with an IPsec SA configured to support the destination and if matched will be processed and IPsec encrypted before being transmitted from the TOE.

FCS_IPSEC_EXT.1.3

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).

Verdict: Pass

Section 7.1.2.9 states that the TOE supports IPsec tunnel-mode.

FCS_IPSEC_EXT.1.4

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

Verdict: Pass

Section 7.1.2.9 states that the TOE supports for ESP protection with AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128, AES-GCM-192 and AES-GCM-256 using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512.

FCS_IPSEC_EXT.1.5

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.

Verdict: Pass

Section 7.1.2.9 states that the TOE implements IKEv2 with:

- IKEv2 as defined in RFCs 5996 (including support for NAT traversal) and RFC 4868 for hash functions.

The TOE supports IKEv2 protection with AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128, AES-GCM-192, and AES-GCM-256.

FCS_IPSEC_EXT.1.6

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.

Verdict: Pass

Section 7.1.2.19 states that the TOE supports IKEv2 protection with AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128, AES-GCM-192, and AES-GCM-256.

FCS_IPSEC_EXT.1.7

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.

Verdict: **Pass**

Section 7.1.2.9 states that the TOE supports configuration of IKE lifetimes as follows. The TOE will initiate SA renegotiation prior to expiration or consumption of these lifetimes.

- IKEv2 SA lifetime by length of time (300 seconds to 24-hours (86400 seconds))
- IKEv2 Child SA lifetimes of number of bytes (1 or unlimited MBytes)
- IKEv2 Child SA lifetimes length of time (300 seconds to 8-hours (28800 seconds)).

FCS_IPSEC_EXT.1.8

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.

Verdict: **Pass**

See FCS_IPSEC_EXT.1.7 above.

FCS_IPSEC_EXT.1.9

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.

Verdict: **Pass**

Section 7.1.2.9 states that the TOE uses AES-CTR DRBG to generate the secret value ‘x’ used in the IKEv2 Diffie-Hellman key exchange (“x” in g^x mod p). These exponents generated for DH Group 14 are 224-bits (28 bytes) long. Nonces generated using AES-CTR DRBG are 256-bits (32-bytes), which is more than 128-bits in size and half of the largest output size supported (SHA-512).

FCS_IPSEC_EXT.1.10

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.

Verdict: Pass

See FCS_IPSEC_EXT.1.9 above.

FCS_IPSEC_EXT.1.11

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.

Verdict: Pass

Section 7.1.2.9 states that the TOE supports Diffie-Hellman Group 14 only.

FCS_IPSEC_EXT.1.12

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.

Verdict: Pass

Please see the response from FCS_IPSEC_EXT.1.1above.

FCS_IPSEC_EXT.1.13

The evaluator ensures that the TSS identifies RSA and/or ECDSA as being used to perform peer authentication. The description must be consistent with the algorithms as specified in FCS_COP.1/SigGen Cryptographic Operations (for cryptographic signature).

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.

Verdict: Pass

Section 7.1.2.9 states that the TOE supports peer authentication with RSA and ECDSA for use with X.509v3 certificates (conformant to RFC 4945) or pre-shared keys for IPsec IKE. Pre-shared keys are manually entered as text-based strings when configuring IPsec tunnels via the CLI or Web/GUI administration interfaces with a length of 1 – 127 characters. Pre-shared keys entered are combinations of lower and upper-case characters, numeric digits, and supported special characters. For a given IPsec peer's IKE configuration using a pre-shared key, the key must be the same on the configured IPsec peer as entered by the TOE administrator.

FCS_IPSEC_EXT.1.14

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.

Verdict: Pass

Section 7.1.2.9 states that the TOE will attempt match an FQDN (Fully Qualified Domain Name) or IP Address value configured by an authorized administrator for the reference identifier of a given IPsec peer with values of the Subject or Subject Alternative Name (SAN, if present) fields in the peer's certificate when establishing an IKE session with the peer. If neither of the fields match the configured reference identifier, the IPsec tunnel establishment is failed, and a corresponding event is logged.

FCS_NTP_EXT.1

TSS

FCS_NTP_EXT.1.1

The evaluator shall examine the TSS to ensure it identifies the version of NTP supported, how it is implemented and what approach the TOE uses to ensure the timestamp it receives from an NTP timeserver (or NTP peer) is from an authenticated source and the integrity of the time has been maintained. The TOE must support at least one of the methods or may use multiple methods, as specified in the SFR element 1.2. The evaluator shall ensure that each method selected in the ST is described in the TSS, including the version of NTP supported in element 1.1, the message digest algorithms used to verify the authenticity of the timestamp and/or the protocols used to ensure integrity of the timestamp.

Verdict: Pass

Section 7.1.2.10 states that the TOE supports NTP v4 (compliant with RFC 5905). The TOE can be configured for three (3) or more remote, NTP servers.

The TOE uses IPsec for trusted channels to communicate with these servers, thereby ensuring that the TOE is using an authentic time source and that integrity of time is maintained.

2.9 FIA: Identification and authentication (selection-based)

FIA_X509_EXT.1/Rev

TSS

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.

Verdict: Pass

Section 7.1.3.6 states that the TOE can use X.509 certificates for IPsec session authentications. The TOE can be configured with the certificates and their corresponding private keys by authorized administrators creating CSRs on the TOE, exporting the CSRs for certificate authority (CA) signing, and subsequently importing the CA-signed certificates to the TOE.

Authorized administrator can also import intermediate and root-CA certificates for the TOE to present in IPsec session establishments. Intermediate and root-CA (trust anchor) certificates can also be imported to the TOE to support IPsec peer authentications where the IPsec peers do not present these certificates. During IPsec session establishment the TOE will evaluate presented certificates first and when not presented will then evaluate locally available certificates corresponding to the IPsec peer's certificate chain.

The TOE will validate presented or locally available certificates during IPsec session establishment. This validation includes checking to ensure that the basicConstraints extension and CA flag are set to TRUE for all CA certificates Revocation status is checked during IPsec session establishment for all certificates in the peer's certificate chain (path) per OCSP or CRL servers indicated in the certificates. If the indicated OCSP server or CRL distribution point are unavailable to determine revocation status, the TOE will assume, by default, the certificate is not revoked. Invalid or revoked certificates detected will cause the IPsec session to fail to establish and with corresponding errors logged accordingly.

The TOE similarly validates certificates when imported to the TOE, albeit without revocation status checks. Certificates that fail this validation will not be included in the TOE's certificate store for use by IPsec and will be left inert until deleted by an authorized administrator.

When generating Certificate Signing Requests (CSRs), an authorized administrator can select the size of the key as 2048 bits for RSA and 256 or 384 bits for ECDSA. In addition to adding the public key to the certificate details, the administrator can provide information for the Common Name, Organization, Organizational Unit, and Country. The administrator can also provide the following additional TOE specific information:

- Locality
- State/Province
- E-Mail Address

When generating CSRs on the TOE, the Organization Unit (OU) information items is prompted with the name “Division”.

FIA_X509_EXT.2

TSS

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.

Verdict: Pass

See FIA_X509_EXT.1/Rev above.

FIA_X509_EXT.3

TSS

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.

Verdict: Pass

See FIA_X509_EXT.1/Rev above.

2.10 FMT: Security management (selection-based)

FMT_MTD.1/CryptoKeys

TSS

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.

Verdict: Pass

Section 7.1.4.5 states that the TOE restricts the ability to manage IPSec private keys to authorized administrators of the TOE. This includes configuring IPsec pre-shared keys and generating/deleting IPsec RSA and ECDSA public and private keys using applicable configuration CLI commands or Web/GUI operations.

FMT_MOF.1/Functions

TSS

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).

Verdict: Pass

Section 7.1.4.5 states that the TOE restricts the ability to alter the configuration for external audit servers (e.g. add, delete, enable, disable, or modify). This ability, using the **logging** CLI command or Web/GUI equivalent operation, is restricted to authorized administrators of the TOE.

3. GUIDANCE DOCUMENTATION ASSURANCE ACTIVITIES

3.1 FAU: Security audit

FAU_GEN.1

Guidance Documentation

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.

Verdict: Pass

Table 5 in A10 Networks Thunder Series Appliances Common Criteria Configuration Guide contains samples for all claimed SFRs and Named Auditable Events under FAU_GEN.1. Below are some of such auditable events and their corresponding records.

AUDITABLE EVENT **SAMPLE RECORD**

Administrative login and logout **(FAU_GEN.1)**	**Console Login** Mar 08 2021 03:07:06 Notice [SYSTEM]:A cli session for user "niap-user1" from 127.0.0.1 has been opened. Session ID assigned is 55. Mar 08 2021 03:07:04 Info [SYSTEM]:Local authentication successful (user: niap-user1). **Remote SSH Login** Mar 08 2021 03:28:56 Notice [SYSTEM]:A cli session for user "niap-user1" from 10.1.1.165 has been opened. Session ID assigned is 60. Mar 08 2021 03:28:54 Info [SYSTEM]:Local authentication successful (user: niap-user1). **Remote GUI Login** Mar 08 2021 03:41:45 Notice [SYSTEM]:A web session for user "niap-user1" from 10.1.1.165 has been opened. Session ID assigned is 64. Mar 08 2021 03:41:45 Info [SYSTEM]:Local authentication successful (user: niap-user1).
Security related configuration	**CLI Configured Login Banners**

AUDITABLE EVENT	SAMPLE RECORD
changes **(FAU_GEN.1)** Configure the access banner **(FMT_SMF.1)**	May 25 2021 01:09:29 Info [SYSTEM]:ACOS Login banner is set. May 25 2021 01:09:29 [admin] cli: [10.65.25.166:53288] banner login "SSH–Test Pre-Login Message" May 25 2021 01:37:48 Info [SYSTEM]:ACOS EXEC banner is set. May 25 2021 01:37:48 [admin] cli: [10.65.25.166:40114] banner exec "SSH–Test Post-Login Message" **GUI Configured Login Banners** May 25 2021 01:54:28 [admin] web: [361:10.65.25.166:13166] payload section 1 {"web-service": {"axapi-idle": 10, "axapi-session-limit": 30, "gui-idle": 15, "gui-session-limit": 30, "port": 80, "secure-port": 443, "login-message": "GUI-Test Post-Login Message", "pre-login-message": "GUI-Test Pre-Login Message."}} May 25 2021 01:54:28 [admin] web: [361:10.65.25.166:13166] PUT: /axapi/v3/web-service
Set the Time Which is Used for Time-stamps **(FMT_SMF.1)**	**Local Clock Time Change** Sep 03 2021 10:10:10 Notice [TM]:Time has been changed, current: Fri Sep 03 10:10:10 PDT 2021, previous: Fri Sep 03 23:36:51 PDT 2021 Sep 03 2021 23:36:51 [admin] cli: [127.0.0.1] clock set 10:10:10 September 3 2021 Sep 03 2021 23:36:35 [admin] cli: [127.0.0.1] timezone America/Los_Angeles

We made a determination of the administrative actions related to TSF data related to configuration changes. We examined the guidance documentation and made a determination of which administrative commands are related to the configuration of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP. We reviewed the Security Target, Table 5 and the rest of the A10 Networks Thunder Series Appliances Common Criteria Configuration Guide to determine which actions in the administrative guide are related to TSF data related to configuration changes.

Administrative Actions: **CLI Function or GUI equivalent**	**Ref'd In AGD**	**Description**
admin	2.2.2, 3.1, 3.6	Config admin accounts
admin-lockout	2.2.2, 3.6	Config failed auth lockout
audit	2.2.2, 3.7.1	Config auditing
banner	2.2.2, 3.4.1	Config CLI pre-login banner (GUI banner is configured via GUI)

logging	2.2.2, 3.7.1, 3.7.2 3.7.3	Configure audit logging/syslog
ntp	3.5.2	Config NTP
pki	3.11.1	Config Key & Cert info
system	2.2.1, 2.2.2, 3.2	Enable/disable FIPS mode, P/W poli
terminal idle-timeout	2.2.2, 3.3.2	Config idle session timeouts
timezone	2.2.2, 3.5.1	Config system time (timezone)
vpn	2.2.3, 3.13.2, 3.13.3,	IPSec configuration
web-service	3.3.2	Config GUI idle timeout
clear audit	3.7.1	Clear audit data
clear logging	3.7.1	Clear logging data
clock set	2.2.2, 3.5.1	Config system time (timezone)
import ca_cert	3.11.4	Import CA Cert
import cert	3.11.3	Import TOE or Intermediate Cert
access-list	2.2.5, 2.2.6, 3.5, 3.7.2, 3.8, 3.10.2 3.12	Config permit/deny rules

TD0563: The date and time information for any audit event shall be recorded as part of each audit record to ensure the timing of the event can be unambiguously determined from the data contained in the audit record. The representation of date and time information recorded for each event needs to allow unanimous determination of at least day, month and year information for the date and hours, minutes and second information for the time.

FAU_GEN.2

Guidance Documentation

(This is already covered by the Guidance Documentation requirements for FAU_GEN.1.)

Verdict: Pass

FAU_STG_EXT.1

Guidance Documentation

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.

Verdict: Pass

Section 3.7.2 "Remote Logging to Syslog" of the CCCG addresses establishment of the trusted channel to the audit server. It also describes requirements on the audit server, as well as configuration of the TOE needed to communicate with the audit server. It should be noted that A10 has no specific requirements for Syslog.

Section 3.7.2 "Remote Logging to Syslog" of the CCCG also addresses relationship between the local audit data and the audit data that are sent to the audit log server. It should be noted that A10 has no specific requirements for Syslog. When configured for logging to one or more Syslog servers, new audit and event records are saved in the local audit and event logging stores and are immediately sent to the Syslog servers via the IPsec encrypted channel.

Section 3.13 "IPsec Tunnel Management" of the CCCG describes IPsec aspect of trusted channel establishment.

Section 3.7.1 "Local Logging" of the CCCG describes all possible configuration options for FAU_STG_EXT.1.3 and the resulting behaviour of the TOE for each possible configuration. In particular, it states: "Audit entries are generated and stored in two logging stores on the TOE, one for ACOS audit category records and one for ACOS event category records. Each store supports a circular log with the oldest records overwritten when the logging store is full."

3.2 FCS: Cryptographic support

FCS_CKM.1

Guidance Documentation

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.

Verdict: Pass

Section 1.3.3 “Elements Excluded from the TOE via Guidance” of the A10 Networks CCCG lists key sizes not to be used.

Section 3.11 “X.509 Certificate and Key Management” lists supported key sizes. The TOE supports the following key sizes for the CC evaluated configuration:

- RSA: 2048 bits
- ECDSA: 256, 384 bits

Section 3.11.1 “Generate Private Key + Certificate Signing Request (CSR), Export CSR” describes 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 3.13.2 “Configure a VPN IKE Gateway” in A10 Networks CCCG describes the configuration of the algorithm selected in the requirement.

In particular, steps 4, 5 and 6 describe configuration of enabling diffie-hellman-group14.

Section 3.13.3 “Configure a VPN IPsec Tunnel” in A10 Networks CCCG also describes the configuration of the algorithm selected in the requirement. In particular, step 3 describes how to enable diffie-hellman-group14.

FCS_CKM.2

Guidance Documentation

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

Verdict: Pass

Section 2.2.3 “Establish an IPsec Tunnel for Connections in the Operational Environment” in the A10 Networks CCCG describes an IPsec tunnel needs to be set-up to support secure access to external servers and from remote management administrators in the TOE’s operational environment. This set up requires the use of dh-group 14.

Section 3.11.1 “Generate Private Key + Certificate Signing Request (CSR), Export CSR” describes 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 3.13.2 “Configure a VPN IKE Gateway” in A10 Networks CCCG describes the configuration of the algorithm selected in the requirement.

In particular, steps 4, 5 and 6 describe configuration of enabling diffie-hellman-group14.

Section 3.13.3 “Configure a VPN IPsec Tunnel” in A10 Networks CCCG also describes the configuration of the algorithm selected in the requirement. In particular, step 3 describes how to enable diffie-hellman-group14.

FCS_CKM.4

Guidance Documentation

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 command3 and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

Verdict: Pass

There are no configurations or circumstances that do not confirm to key destruction in the TSS. ACOS has no key destruction delays and there are no storage or garbage collection delays for key destruction in ACOS.

Section 3.11.1 “Generate Private Key + Certificate Signing Request (CSR), Export CSR” of the A10 Networks CCCG has instructions on how to delete keys and notes that such key deletion is immediate.

FCS_COP.1/DataEncryption

Guidance Documentation

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.

Verdict: Pass

Section 3.13.1 “IPsec Supported on the TOE” and section 3.13 “IPsec Tunnel Management” of the A10 Networks CCCG address configuration of the key size for IPsec for supported mode (e.g. CBC mode).

IKEv2 support the following algorithms for Encryption: AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128, AES-GCM-192 and AES-GCM-256.

ESP supports the following algorithms for Encryption: AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128, AES-GCM-192 and AES-GCM-256.

Section 3.13.2 “Configure a VPN IKE Gateway” of the A10 Networks CCCG describes how ACOS VPN IKE Gateway supports configuration for IKE on the TOE for a tunnel to a given VPN peer and how they are configured using the vpn ike-gateway CLI command. (e. g. IKE configuration step 7 allows choosing of encryption algorithm).

Section 3.13.3 “Configure a VPN IPsec Tunnel” of the A10 Networks CCCG describes support of configuration for ESP on the TOE and how they are configured using the vpn IPsec CLI command. (e. g. ESP configuration step 4 allows choosing of encryption algorithm.)

- NOTE: If the key size of the encryption setting or the VPN IPsec Gateway instance is greater than the key size for the VPN IKE Gateway instance, the TOE will not instantiate the tunnel with the VPN peer and will log a corresponding error.

FCS_COP.1/SigGen

Guidance Documentation

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.

Verdict: Pass

Section 3.13.2 “Configure a VPN IKE Gateway” of the A10 Networks CCCG describes how to configure IKE in steps 4, 5, 6 to choose either RSA or ECDSA.

In particular, step 4 says: “Choose the authentication method to be used for IKE Phase 1. If use of a Pre-Shared Key is selected, ike-psk-string is the 1 - 127 character string text of the key. Alternately, RSA or ECDSA methods may be chosen.

```
ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# auth-method preshare-key ike-psk-string
```

or

```
ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# auth-method rsa-signature
```

or

ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# auth-method ecdsa-signature.”

Step 5 says: “(For RSA Method) If RSA method selected, add an X.509 certificate and corresponding RSA private key and enable diffie-hellman-group14 for the instance. See Section 3.11 for discussion on configuring X.509 certificates and keys.

```
ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# local-cert rsa-cert1-signed
ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# key  rsa-cert1
ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# dh-group 14
```

- NOTE: this example uses the RSA key and X.509 signed certificate configured from examples in Section 3.11.

Step 6 says: "(For ECDSA Method) If ECDSA method selected, add an X.509 certificate and corresponding ECDSA private key and enable diffie-hellman-group14 for the instance. See Section 3.11 for discussion on configuring X.509 certificates and keys.

```
ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# local-cert ecdsa-cert1-signed
ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# key ecdsa-cert1
ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# dh-group 14
```

- NOTE: this example uses the ECDSA key and X.509 signed certificate configured from examples in Section 3.11. "

FCS_COP.1/Hash

Guidance Documentation

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

Verdict: Pass

Section 3.13.1 "IPsec Supported on the TOE" and section 3.13 "IPsec Tunnel Management" of the A10 Networks CCCG provide information regarding configuring the required hash sizes (in IPsec).

Section 3.13.2 "Configure a VPN IKE Gateway" of the A10 Networks CCCG describes how IKE configures hash sizes are configured.

```
ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# encryption aes-256 hash sha256
```

Section 3.13.3 "Configure a VPN IPsec Tunnel" of the A10 Networks CCCG describes how IPsec hash sizes are configured. In particular, it says: "Choose encryption and integrity algorithms for the instance.

```
ACOS(config-ipsec:ipsec_mgmt_esp_1)# encryption aes-256 hash sha256
```

"

The SHA algorithm that is used with the RSA digital signature of the upgrade image is SHA-256. It is not configurable.

FCS_COP.1/KeyedHash

Guidance Documentation

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.

Verdict: **Pass**

Section 3.13.1 “IPsec Supported on the TOE” and section 3.13 “IPsec Tunnel Management” of the A10 Networks CCCG provide information regarding configuring the required HMAC values (in IPsec).

Section 3.13.2 “Configure a VPN IKE Gateway” of the A10 Networks CCCG describes how IKE HMAC values are configured.

```
ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# encryption aes-256 hash sha256
```

Section 3.13.3 “Configure a VPN IPsec Tunnel” of the A10 Networks CCCG describes how IPsec HMAC values are configured.

```
ACOS(config-ipsec:ipsec_mgmt_esp_1)# encryption aes-256 hash sha256
```

FCS_RBG_EXT.1

Guidance Documentation

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

Verdict: **Pass**

No specific instructions are needed to configuring the RNG functionality. RNG is inherent in the TOE and not subject to configuration by TOE ADMINs.

3.3 FIA: Identification and authentication

FIA_AFL.1

Guidance Documentation

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.

Verdict: Pass

Section 3.6 “Failed Authentication Lockout” in A10 Networks CCCG describes instructions for configuring the number of successive unsuccessful authentication attempts and time period, and the process of allowing the remote administrator to once again successfully log on is described for each “action” specified.

```
ACOS(config)# admin-lockout threshold   3                    --- default is  5 attempts
ACOS(config)# admin-lockout duration   15                    --- default is 10 minutes
```

```
ACOS(config-admin:adminuser1)# unlock
```

Section 3.6 “Failed Authentication Lockout” in A10 Networks CCCG also 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.

Locked administrators accounts are only blocked for remote CLI and Web/GUI management access. These accounts are not blocked during lockout for administrators able to access to the local console of the TOE.

To maintain compliance with the TOE CC evaluated configuration, ensure that the lockout attempts threshold (**admin-lockout threshold**) is configured for a non-zero value.

FIA_PMG_EXT.1

Guidance Documentation

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.

Verdict: Pass

Section 3.2 “Password Management” in A10 Networks CCCG describes the characters that may be used in passwords and provides guidance to security administrators on the composition of strong passwords.

This section also provides instructions on setting the minimum password length and describes the valid minimum password lengths supported.

The password policy is configurable by TOE administrators and supports the minimum password length of 8 characters and a maximum password length of 63 characters.

FIA_UIA_EXT.1

Guidance Documentation

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.

Verdict: Pass

Section 2 “Secure Installation and Configuration” of the A10 Networks CCCG describes initial set up the TOE for base security factors.

Section 3 “Secure Management” of the A10 Networks CCCG describes additional factors for configuring the TOE, including certificates, tunnels, login banners, adding admins, etc.

FIA_UAU_EXT.2

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.

FIA_UAU.7

Guidance Documentation

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.

Verdict: Pass

Section 3.2 “Password Management” in A10 Networks CCCG states that password information is never revealed during interactive administrator logins to the TOE. For logins using the CLI to the local console, input characters are simply not echoed. Preparation is not mentioned, since no preparation is needed.

3.4 FMT: Security management

FMT_MOF.1(1)/ManualUpdate

Guidance Documentation

Verdict: **Pass**

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).

Section 3.9 “Trusted Updates” in A10 Networks CCCG describes necessary steps to perform manual update.

The **upgrade** CLI command can be used to update the TOE

If the connection fails to establish or is lost while downloading the TOE update image from the trusted server, the update operation will be failed, and the administrator will have to retry the operation at a later time when reliable access to the server(s) is available.

FMT_MTD.1/CoreData

Guidance Documentation

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.

Verdict: **Pass**

- Section 3.1 of the A10 Networks CCCG explains that all read-write administrators are considered Security Administrators of the TOE.

The following lists the TOE security management functions claimed by ST and where in the A10 Networks CCCG the usage of these functions is described:

- Ability to administer the TOE locally and remotely - Section 1.3.4 (for scope of local/remote management), and Section 3.1 (for description of how to configure accounts for local/remote administration).

- Ability to configure the access banner – Section 3.4.

- Ability to configure the session inactivity time before session termination or locking – Section 3.3.

- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates - Section 3.9 (for updating).

- Ability to configure the authentication failure parameters for FIA_AFL.1- Section 3.6.

- Ability to configure audit behavior (e.g. changes to storage locations for audit; changes to behavior when local audit storage space is full) – Section 3.7.

- Ability to manage the cryptographic keys – Section 3.11.1.

- Ability to configure the cryptographic functionality – Section 3.13.

- Ability to configure the lifetime for IPsec SAs – Section 3.13.2 (for IKEv2 SA lifetimes), Section 3.13.3 (for IKEv2 Child SA lifetimes).

- Ability to re-enable an Administrator account – Section 3.10.2.

- Ability to set the time which is used for time-stamps – Section 2.2.2.

- Ability to configure NTP – Section 3.5.

- Ability to configure the reference identifier for the peer – Section 2.2.3 and Section 3.13.2 (step 9)

- Ability to import X.509v3 certificates to the TOE's trust store – Section 3.11.3 (for imported signed certificates), Section 3.11.4 (for importing Root CA certificates).

FMT_SMF.1

Guidance Documentation

Verdict: Pass

Addressed in the section above.

FMT_SMR.2

Guidance Documentation

Verdict: Pass

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 3.10.1 “SSH Clients via IPsec” and section 3.10.2 “Web/GUI Clients via IPsec” in A10 Networks CCCG contain instructions for administering the TOE remotely, including any configuration that needs to be performed on the client for remote administration.

Section 2.2.1 “Initial Connection and FIPS Mode Confirmation” of the A10 Networks CCCG describes connecting to the TOE via the local console using a terminal application.

3.5 FPT: Protection of the TSF

FPT_SKP_EXT.1

Guidance Documentation N/A

FPT_APW_EXT.1

Guidance Documentation N/A

FPT_TST_EXT.1

Guidance Documentation

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.

Verdict: Pass

Section 3.14 “Boot Time Integrity Self-Tests” in A10 Networks CCCG states that during boot-up from either a power-on or a reboot, the TOE will perform integrity checks on the TOE software and TOE cryptographic capabilities. If any of these tests fail the TOE is put into FIPS failure mode (A10 ACOS specific state). These possible errors correspond to those described in the TSS.

The FIPS failure mode is indicated by the following:

- The prompt: ACOS (FIPS FAIL MODE) #
- Integrity check: Image verification check failed / FIPS Power On Self-Test failed
- Failure of the TOE's cryptographic capabilities: FIPS library power on self-test failed / FIPS Power On Self-Test failed.

If this condition occurs, the following actions should be taken:

- Power-cycle and restart the TOE to perform these tests again and determine if normal operation can be resumed

- If this condition persists, contact A10 Networks support to troubleshoot the issue and coordinate replacement of the hardware, if needed.

FPT_TUD_EXT.1

Guidance Documentation

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.

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.

Verdict: Pass

Section 3.9 "Trusted Updates" in A10 Networks CCCG describes how to query the currently active version. In particular, it says: "The **show version** CLI command can be used to show the software versions installed on the TOE's primary and secondary boot-images. The image indicated with "(default)" is the boot-image currently enabled for the TOE."

Section 3.9 also provides description of how to query the loaded but inactive version, noting that if admin doesn't want to reboot right away admin can do so at a later time. In particular, it says: "If the TOE update image had just been installed for the active (running) boot-image, the administrator may choose to reboot the device immediately, as prompted by the upgrade CLI command or at a later time of the administrator's discretion when the update to take effect."

Section 3.9 "Trusted Updates" describes how the verification of the authenticity of the update is performed (digital signature verification). In particular, it says: "The TOE update image downloaded is an ".upg" file, signed with a digital signature. Before installing the TOE update image, the signature is verified against a key stored in the TOE. If the digital signature verifies successfully, an MD5 integrity check will additionally be performed for the contents of the update image. If the digital signature and MD5 integrity checks succeed, the update will proceed to be installed… If any of these tests fail, the TOE update operation will be failed with visual feedback to the administrator initiating the update, along with corresponding logged failure events."

FPT_STM_EXT.1

Guidance Documentation

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.

Verdict: Pass

Section 3.5.1 "System Time Configuration" and section 2.2.2 "Basic System Parameter Configuration" of the A10 Networks CCCG provides instructions for the administrator on how to set the time. In particular, it says: "The TOE maintains date/time based on the system clock provided by its underlying hardware. This system can be configured (set) by the administrator using the clock and timezone CLI commands, previously described in Section 2.2.2."

Section 3.5.2 "Network Time (NTP) Configuration" of the A10 Networks CCCG provides instructions on 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. In particular, it says: "The TOE supports up to three (3) NTP servers to synchronize the TOE with the network date and time with communication is protected by the IPsec tunnel in the CC evaluated configuration."
NTP servers can be configured using the **ntp server** CLI command.

Section 2.2.3 "Establish an IPsec Tunnel for Connections in the Operational Environment" of the CCCG describes establishment of IPsec Tunnel.

3.6 FTA: TOE Access

FTA_SSL_EXT.1

Guidance Documentation

Verdict: Pass

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 3.3.2 "Inactivity Termination" of the A10 Networks CCCG provides instructions for configuring the inactivity time period. In particular, it says: "Authenticated sessions to the TOE can be configured for inactivity timeouts up to 60 minutes using the **terminal idle-timeout** CLI command". This timeout will affect management sessions to the TOE CLI on the local console. The default for this idle-time out is 15 minutes."

"To maintain compliance with the TOE CC evaluated configuration, one must ensure that these timeout settings are configured for non-zero values."

FTA_SSL.3

Guidance Documentation

Verdict: Pass

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

Section 3.3.2 "Inactivity Termination" of the A10 Networks CCCG provides instructions for configuring the inactivity time period for remote administrative session termination.

In particular, it says: "Authenticated sessions to the TOE can be configured for inactivity timeouts up to 60 minutes using the **terminal idle-timeout** CLI command. This timeout will affect management sessions to the TOE CLI on the local console or remotely using SSHv2 via IPsec. The default for this idle-time out is 15 minutes. For example:

```
ACOS(config)# terminal idle-timeout 10
```

Separate web service timeout values can be similarly configured to control inactivity timeouts for authenticated sessions to the TOE's web GUI via IPsec. These timeouts are configured by the

web-service gui-timeout-policy idle CLI command. The default for this idle-time out is 10 minutes. For example:

```
ACOS(config)# web-service gui-timeout-policy idle 15
```

To maintain compliance with the TOE CC evaluated configuration, one must ensure that these timeout settings are configured for non-zero values."

FTA_SSL.4

Guidance Documentation

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

Verdict: Pass

Section 3.3.1 “Graceful Termination” of the A10 Networks CCCG provides information on how to terminate a local or remote interactive session. In particular, it says: “The TOE allows interactive sessions to exit (logout) gracefully by command (operation). Administrative sessions to the TOE CLI using the local console or remotely accessing the CLI with SSHv2 via IPsec can terminate their sessions using one or more instances of the ‘exit’ CLI command. Administrative sessions to the TOE Web/GUI via IPsec can terminate their sessions using the provided ‘logout’ option in the user control menu at the top right corner of the browser window.”

FTA_TAB.1

Guidance Documentation

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

Verdict: Pass

Section 3.4 “Login Banners” of the A10 Networks CCCG describes how to configure the banner message.

Section 3.4.1 “CLI Login Banner” describes how to configure the CLI banner message. This banner is configured for the TOE using the **banner login** CLI command.

Section 3.4.2 “Web/GUI Login Banner” describes how to configure the Web/GUI banner message, This banner is configured for the TOE on **‘System -> Settings -> Web’** Web/GUI page for the parameter **‘Pre-GUI Login Message’**.

3.7 FTP: Trusted path/channels

FTP_ITC.1

Guidance Documentation

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.

Verdict: Pass

Section 2.2.3 “Establish an IPsec Tunnel for Connections” in A10 Networks CCCG describes establishment of an IPsec Tunnel.

Section 3.5.2 “Network Time (NTP) Configuration”, section 3.7.2 “Remote Logging to Syslog”, and section “3.8 File Servers” in A10 Networks CCCG also contain instructions for establishing the allowed protocols with each authorized IT entity.

Section 3.5.2 in A10 Networks CCCG states that the TOE supports up to three (3) NTP servers. If connectivity to a synchronized NTP server is lost, the TOE will attempt to connect with the other configured servers to maintain synchrony with the network time domain.

Section 3.7.2 “Remote Logging to Syslog” in A10 Networks CCCG states that: multiple Syslog servers can be configured on the TOE. If communications with a Syslog server is lost and becomes re-established, new logging records will be logged successfully on the server.

Section 3.9 “Trusted Updates” in A10 Networks CCCG states that if the connection fails to establish or is lost while downloading the TOE update image from the trusted server, the update operation will be failed, and the administrator will have to retry the operation at a later time when reliable access to the server(s) is available.

Section 3.13.2 “Configure a VPN IKE Gateway” of the A10 Networks CCCG describes how ACOS VPN IKE Gateway supports configuration for IKE on the TOE for a tunnel to a given VPN peer and how they are configured using the vpn ike-gateway CLI command.

Section 3.13.3 “Configure a VPN IPsec Tunnel” of the A10 Networks CCCG describes support of configuration for ESP on the TOE and how they are configured using the vpn IPsec CLI command.

FTP_TRP.1/Admin

Guidance Documentation

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

Verdict: Pass

Section 3.10.1 “SSH Clients via IPsec” and section 3.10.2 “Web/GUI Clients via IPsec” in A10 Networks CCCG contain instructions for establishing the remote administrative sessions for each supported method.

Section 3.13.2 “Configure a VPN IKE Gateway” of the A10 Networks CCCG describes how ACOS VPN IKE Gateway supports configuration for IKE on the TOE for a tunnel to a given VPN peer and how they are configured using the vpn ike-gateway CLI command.

Section 3.13.3 “Configure a VPN IPsec Tunnel” of the A10 Networks CCCG describes support of configuration for ESP on the TOE and how they are configured using the vpn IPsec CLI command.

3.8 FCS: Cryptographic support (selection-based)

FCS_IPSEC_EXT.1

Guidance Documentation

FCS_IPSEC_EXT.1.1

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.

Verdict: Pass

The above requirements are addressed in the following sections in A10 Networks CCCG: section 2.2.5 "Restrict Management Traffic to Ping/Traceroute and IPsec Tunnel", section 3.5.2 "Network Time (NTP) Configuration", section "3.7.2 Remote Logging to Syslog", section 3.8 "File Servers", section 3.10.1 "SSH Clients via IPsec", section "3.10.2 Web/GUI Clients via IPsec", and section 3.12 "Certificate Revocation".

Section 3.13.4 "ACL Rules" in A10 Networks CCCG states: "ACLs that allow traffic other than via IPsec peers are effectively rules that "bypass" secure communications in the CC evaluated configuration. Other than rules recommended for ICMP in Section 2.2.5 to support ping and traceroute for network availability detection on the locally connected network, such bypass rules should not be configured on the TOE."

Section 3.13.4 also says: "The administrator is expected to configure a "default deny" rule as the final ACL rule on the management interfaces to ensure that the TOE rejects traffic on the TOE management interface that is not explicitly allowed by earlier rules. This rule is described in Section 2.2.5 "Restrict Management Traffic to Ping/Traceroute and IPsec Tunnel"."

Section 3.13.4 also says: "The third parameter of an ACL is a sequence number to indicate the precedence of evaluation of an ACL relative to others. ACLs with lower valued sequences numbers are evaluated first. The first ACL with a matching rule during an evaluation will apply its indicated action for packet of the evaluation."

The description in the guidance documentation is consistent with the description in the TSS.

FCS_IPSEC_EXT.1.3

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

Verdict: Pass

Section 3.13.1 “IPsec Supported on the TOE” in A10 Networks CCCG states that Tunnel mode is supported only. Transport mode is not supported.

Section 3.13.3 “Configure a VPN IPsec Tunnel” in A10 Networks CCCG contains instructions on how to configure the connection in the tunnel mode selected.

ACOS(config-ipsec:ipsec_mgmt_esp_1)# mode tunnel

FCS_IPSEC_EXT.1.4

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

Verdict: Pass

Section 3.13.1 “IPsec Supported on the TOE” in A10 Networks CCCG states:

- ESP supports the following algorithms and Security Association (SA) lifetimes.
  - Encryption: AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128, AES-GCM-192, AES-GCM-256
  - Integrity: SHA-1, SHA-128, SHA-256, SHA-384

Section 3.13.3 “Configure a VPN IPsec Tunnel” in A10 Networks CCCG describes how to configure the TOE to use the algorithms selected.

ACOS(config-ipsec:ipsec_mgmt_esp_1)# encryption aes-256 hash sha256

or

ACOS(config-ipsec:ipsec_mgmt_esp_1)# encryption aes-gcm-256

FCS_IPSEC_EXT.1.5

The evaluator shall check the guidance documentation to ensure it instructs the administrator how to configure the TOE to use IKEv1 and/or IKEv2 (as selected), and how to configure the TOE to perform NAT traversal (if selected). 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.

Verdict: **Pass**

Section 3.13.2 "Configure a VPN IKE Gateway" in A10 Networks CCCG describes steps for configuring IKEv2 (as selected) and NAT traversal. In particular, step 2 describes configuration for IKEv2 and step 11 describes configuration for NAT traversal.

Section 1.3.3 "Elements Excluded from the TOE via Guidance" states that IKEv1 is not used by the TOE.

FCS_IPSEC_EXT.1.6

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

Verdict: **Pass**

Section 1.3.3 "Elements Excluded from the TOE via Guidance" describes excluded IPsec/IKE configuration settings, such as Encryption Algorithms: DES, 3DES, Null (no encryption) and Hashing Algorithms: MD5, Null (no hash).

Section 3.13.2 "Configure a VPN IKE Gateway" in A10 Networks CCCG describe the configuration of all selected algorithms in the requirement. In particular, step 7 states: "Choose encryption and integrity algorithms for the instance…

ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# encryption aes-256 hash sha256

or

ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# encryption aes-gcm-256

FCS_IPSEC_EXT.1.7

[TD0633] 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.

Verdict: Pass

Section 3.13.2 “Configure a VPN IKE Gateway” in A10 Networks CCCG contains sufficient information so the Administrator is able to configure the values for SA lifetimes. In particular, step 10 states: “Configure the lifetime for Security Association (SA) re-keying on the instance. This is the duration (in seconds) prior to which the TOE will initiate re-keying of the IKE SA.

ACOS(config-ike-gateway:ipsec_mgmt_ike_1)# lifetime 86400.”

FCS_IPSEC_EXT.1.8

[TD0633] The evaluator shall verify that the values for SA lifetimes can be configured and that the instructions for doing so are located in the guidance documentation. If time-based limits are supported, configuring the limit may lead to a rekey no later than the specified limit. For some implementations, it may be necessary, though, to configure the TOE with a lower time value to ensure a rekey is performed before the maximum SA lifetime of 8 hours is exceeded (e.g. configure a time value of 7h 45min to ensure the actual rekey is performed no later than 8h). The evaluator shall verify that the guidance documentation allows the Administrator to configure the Phase 2 SA value of 8 hours or provides sufficient instruction about the time value to configure to ensure the rekey is performed no later than the maximum SA lifetime of 8 hours. It is not permitted to configure a value of 8 hours if that leads to an actual rekey after more than 8hours. Currently there are no values mandated for the number of bytes, the evaluator just ensures that this can be configured if selected in the requirement.

Verdict: Pass

Section 3.13.3 “Configure a VPN IPsec Tunnel” in A10 Networks CCCG describes how to configure the values for SA lifetimes.

In particular, step 5 says: “Configure the time-based lifetime for Security Association (SA) re-keying on the instance. This is the duration (in seconds) prior to which the TOE will initiate renegotiation of the ESP SA.

ACOS(config-ipsec:ipsec_mgmt_esp_1)# lifetime 28800”

and step 6 says: “Configure the traffic volume lifetime for Security Association (SA) re-keying on the instance. This is the duration (in seconds) prior to which the TOE will initiate re-keying of the ESP SA.

ACOS(config-ipsec:ipsec_mgmt_esp_1)# lifebytes 10240.”

FCS_IPSEC_EXT.1.11

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

Verdict: Pass

Section 3.13.2 “Configure a VPN IKE Gateway” in A10 Networks CCCG describes the configuration of the algorithm selected in the requirement.

In particular, steps 4, 5 and 6 describe configuration of diffie-hellman-group14.

Section 3.13.3 “Configure a VPN IPsec Tunnel” in A10 Networks CCCG also describes the configuration of the algorithm selected in the requirement. In particular, step 3 describes how to enable diffie-hellman-group14.

FCS_IPSEC_EXT.1.13

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”.

Verdict: Pass

Steps 5 and 6 of section 3.13.2 “Configure a VPN IKE Gateway” in A10 Networks CCCG describe the configuration of the TOE to use certificates with RSA and ECDSA signatures.

Step 4 of section 3.13.2 “Configure a VPN IKE Gateway” in A10 Networks CCCG describes the establishment of pre-shared keys.

Information regarding how to configure the TOE to connect to a trusted CA and ensure that a valid certificate for that CA is loaded into the TOE and marked “trusted” is contained in the following sections of the A10 Networks CCCG: section 3.11.3 “Import Signed Certificates”, section 3.11.4 “Import Root CA Certificates”, and section 3.13.2 “Configure a VPN IKE Gateway” (steps 5 and 6).

FCS_IPSEC_EXT.1.14

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.

Verdict: Pass

Section 3.13.1 “IPsec Supported on the TOE” in A10 Networks CCCG states that “'reference identifier' value is supported either in Subject or SAN (if present) fields of a received certificate and must contain the FQDN or IP address of the IPsec entity”.

Step 9 of Section 3.13.2 “Configure a VPN IKE Gateway” in A10 Networks CCCG describes how to configure the reference identifier(s) used to check the identity of peer(s).

Section 3.13.2 “Configure a VPN IKE Gateway” in A10 Networks CCCG, after (last) step 11, states that: “an IKEv2 session is a unique combination of the TOE’s management IP address, remote peer’s IP address, and a configured VPN IKE Gateway instance. The VPN IKE Gateway instance additionally determines the local and remote identifiers of the combination, all of which need to be matched along with other compatible settings for the instance, in order for the session to successfully instantiate. For TOE initiated IKEv2 sessions, the remote IPsec peer will perform the matching based on the peer’s configuration and will accept (reject) the session accordingly. For IPsec peer initiated IKEv2 sessions, the TOE will perform this matching.

Guidance Documentation

FCS_NTP_EXT.1.1

The evaluator shall examine the guidance documentation to ensure it provides the Security Administrator instructions as how to configure the version of NTP supported, how to configure multiple NTP servers for the TOE’s time source and how to configure the TOE to use the method(s) that are selected in the ST.

Verdict: Pass

Section 3.5.2 “Network Time (NTP) Configuration” in A10 Networks CCCG provides the Security Administrator with instructions regarding the version of NTP supported, and also describes how to configure multiple NTP servers for the TOE’s time source and how to configure the TOE to use the method that is selected in the ST. the NTP version is not configurable and only NTP v4 is supported. NTP servers can be configured using the **ntp server** CLI command.

Section 2.2.3 “Establish an IPsec Tunnel for Connections in the Operational Environment” of the CCCG describes establishment of IPsec Tunnel.

FCS_NTP_EXT.1.2

For each of the secondary selections made in the ST, the evaluator shall examine the guidance document to ensure it instructs the Security Administrator how to configure the TOE to use the algorithms that support the authenticity of the timestamp and/or how to configure the TOE to use the protocols that ensure the integrity of the timestamp.

Assurance Activity Note:

Each primary selection in the SFR contains selections that specify a cryptographic algorithm or cryptographic protocol. For each of these secondary selections made in the ST, the evaluator shall examine the guidance documentation to ensure that the documentation instructs the Security Administrator how to configure the TOE to use the chosen option(s).

Verdict: Pass

Section 2.2.3 “Establish an IPsec Tunnel for Connections in the Operational Environment” in the A10 Networks CCCG describes that an IPsec tunnel needs to be set-up to support secure access to external servers.

Section 3.5.2 “Network Time (NTP) Configuration in A10 Networks CCCG” further states that NTP communications are secured by IPsec tunnels in the CC evaluated configuration.

FCS_NTP_EXT.1.3

The evaluator shall examine the guidance documentation to ensure it provides the Security Administrator instructions as how to configure the TOE to not accept broadcast and multicast NTP packets that would result in the timestamp being updated.

Verdict: Pass

Section 3.5.2 “Network Time (NTP) Configuration” in A10 Networks CCCG clarifies that: “the TOE does not support NTP broadcast and multicast time updates.”

3.9 FIA: Identification and authentication (selection-based)

FIA_X509_EXT.1/Rev

Guidance Documentation

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.

Verdict: Pass

Section 3.13.2 “Configure a VPN IKE Gateway” states: “Validation of X.509 leaf, intermediate, and root-CA certificates used for IKEv2 are performed immediately after they are imported (transferred) to the TOE. Successfully validated certificates are added to the TOE’s local, trusted IKEv2 certificates store. Certificates that fail to validate are logged and left inert (unused) until they are deleted by a Security Administrator. When establishing IPsec tunnels, the TOE validates certificates received from the IPsec peer and confirms that the IPsec peer certificate and the certificate path are valid (not revoked) as described in Section 3.12 above.”

Section 3.12 “Certificate Revocation” in A10 Networks CCCG describes how certificate revocation checking is performed and on which certificate. In particular it says: “The TOE uses the Online Certificate Status Protocol (OCSP) and Certificate Revocation Lists (CRLs) during validation of peer and local certificates. When a certificate is received from an IPsec peer, the TOE processes the certificate chain path until the last certificate is reached.”

FIA_X509_EXT.2

Guidance Documentation

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.

Verdict: Pass

Section 3.11.1 “Generate Private Key + Certificate Signing Request (CSR), Export CSR” and section

3.11.2 “Generate Certificates Signed by the CA”, and section 3.11.3 “Import Signed Certificates” in A10 Networks CCCG describe the configuration required in the operating environment so the TOE can use the certificates.

Section 3.13.2 “Configure a VPN IKE Gateway” in A10 Networks CCCG contains required configuration on the TOE to use the certificates. In particular, steps 5 and 6 are described as following:

5. (For RSA Method) If RSA method selected, add an X.509 certificate and corresponding RSA private key…for the instance. See Section 3.11 for discussion on configuring X.509 certificates and keys…

6. (For ECDSA Method) If ECDSA method selected, add an X.509 certificate and corresponding ECDSA private key… for the instance. See Section 3.11 for discussion on configuring X.509 certificates and keys…

Section 3.12 “Certificate Revocation” in A10 Networks CCCG states that “On occasions where the referenced OCSP servers or CRL distribution points are unavailable, the certificate being validated will be not revoked.”

FIA_X509_EXT.3

Guidance Documentation

The evaluator shall check to ensure that the guidance documentation contains instructions on requesting certificates from a CA, including generation of a Certificate Request. If the ST author selects "Common Name", "Organization", "Organizational Unit", or "Country", the evaluator shall ensure that this guidance includes instructions for establishing these fields before creating the Certification Request.

Verdict: Pass

Section 3.11.1 “Generate Private Key + Certificate Signing Request (CSR), Export CSR” and section

3.11.2 “Generate Certificates Signed by the CA” in A10 Networks CCCG contain instructions on requesting certificates from a CA, including generation of a Certificate Request.

Also, section 3.11.1 “Generate Private Key + Certificate Signing Request (CSR), Export CSR” in A10 Networks CCCG contains instructions for establishing these fields before creating the Certification Request.

3.10 FMT: Security management (selection-based)

FMT_MOF.1/Functions

Guidance Documentation

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.

Verdict: Pass

Section 3.7.2 “Remote Logging to Syslog” of the CCCG addresses establishment of the trusted channel to the audit server. It also describes requirements on the audit server, as well as configuration of the TOE needed to communicate with the audit server. It should be noted that A10 has no specific requirements for Syslog. Multiple Syslog servers can be configured on the TOE by using the **logging host** CLI command.

FMT_MTD.1/CryptoKeys

Guidance Documentation

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.

Verdict: Pass

Section 3.11 “X.509 Certificate and Key Management” in A10 Networks CCCG lists the keys the Security Administrator is able to manage to include the options available and how those operations are performed.

4. TEST ASSURANCE ACTIVITIES

4.1 TEST EQUIVALENCE JUSTIFICATION

The following equivalency analysis provides an analysis of key areas of differentiation of the hardware model to determine the minimum subset to be used in testing. The areas examined and analysis description provided consistent with those specified in Section A.7 of the supporting documentation for the NDcPP v2.2e.

The model specific hardware and TOE configurations are shown in Table 4 of the ST and, in summary, include the following models.

- TOE Hardware Models : TH-4435, TH-5840-11, TH-7445, TH-7650-11, TH-7655

Platform/Hardware Dependencies

The TOE boundary is inclusive of all hardware required by the TOE. The hardware models do not provide any of the TSF functionality. All security functionality is implemented in model independent code that is identical across hardware models. For these networking appliances, the hardware within the TOE only differs by configuration and performance, such as networking bandwidth/ports and CPU processing speed. Furthermore, entropy sources for the TOE, in general, and specifically as used to seed the software DRBG implemented within ACOS are identical across TOE models. There are no hardware specific dependencies of the product.

- Result: All models are equivalent.

Differences in TOE Software Binaries

This category of differences is only applicable if the TOE is installed on an OS outside of the TOE boundary. In this case, all software including the OS is included in ACOS and hence within the TOE boundary. The software for all models comprising the TOE is ACOS 5.2.1-P3 and is identical for all the TOE models. There are no specific dependencies on the OS since the TOE will not be installed on different OSs.

- Result: All models are equivalent.

Differences in Libraries Used to Provide TOE Functionality

All software binaries compiled in the TOE software are identical and have the same version numbers.

There are no differences between the libraries included in the software binaries compiled for the TOE software, simply as there is only a single binary. It is worth noting that this includes the libraries the TOE uses to provide its cryptographic functionality which are equally the same for all TOE models. There is no difference in libraries for any model.

- Result: All models are equivalent.

TOE Management Interface Differences

The TOE is managed either remotely through CLI or Web/GUI sessions via IPsec or locally with the CLI on the console of the device. These management options are available on all TOE models regardless of the configuration. There is no effective difference in the TOE ethernet management interfaces between the TOE models as the standard Linux kernel drivers involved (e1000e and i40e) were both developed by Intel.

- Result: All models are equivalent.

TOE Functional Differences

Each model within the TOE boundary provides identical functionality. There is no difference in the way the users interact with the device or services are available for each of these models. There is no difference in the way the device or services interact with other elements in the TOE operational environment (e.g. authentication servers, syslog servers, etc) for each of these models. There are no functional differences between the various TOE models.

- Result: All models are equivalent.

Differences in Libraries Use of Intel ISA Extensions Between CPU Models

As described below in Section 1.2.6.7, the only difference between the CPU variants across the target models during run-time of the common code images in the TOE (see Section 1.2.6.3) affecting cryptographic operations are the support for AVX/AVX2/AVX-512 extensions to the Intel Instruction Set. Only the OpenSSL and Linux Kernel software components of the TOE vary their execution based on the AVX implementation supported on the operational CPU.

These run-time differences are effectively exercised and validated on their respective CPU variants per the NIST CAVP/AVP certifications indicated in Table 6 of the Security Target (ST) document for these CPU variants. No effective differences between the various TOE models for NIAP testing scope.

- Result: All models are equivalent.

OS and Processor Comparison

The following tables compare the Operating System and CPU within each of the included TOE models.

TOE Operating Systems

TOE MODEL	OPERATING SYSTEM	ANALYSIS
TH-4435 TH-5840-11, TH-7445, TH-7650-11, TH-7655	ACOS 5.2.1-P3	All models are running the exact same version of ACOS. There are NO differences.

TOE CPUs

TOE MODEL	PROCESSOR	ANALYSIS
TH-4435	1x Intel Xeon E5-2680v2	All models are running an Intel Xeon CPUs that support the Intel Secure Key feature for Digital Random Number Generation (DRNG), including support for both RDRAND and RDSEED instructions. The three CPU variants do differ in ISA extension support, most notably in their support, or lack thereof, for AVX2 and AVX-512. All three CPU variants support AVX. Intel defines instruction set extensions as “additional instructions which can increase performance when the same operations are performed on multiple data objects”. Based on this definition the CCTL has not concluded that these would interfere with SFR functionality in any meaningful way. Additionally, all CPU variants have been validated as indicated in Table 6 of the ST. Based on the above it is concluded that these differences would not interfere with SFR functionality in any meaningful way.
TH-5840-11, TH-7445	1x Intel Xeon E5-2695v4 2x Intel Xeon E5-2695v4	
TH-7650-11, TH-7655	2x Intel Xeon Gold 6258R	

Recommendations/Conclusions

Based on the equivalency rationale above, testing will be performed on the following subset of TOE models:

- TH-4435

4.2 TESTS

4.2.1 Security Audit (FAU)

Note: as this TOE is not distributed, none of the security functional requirements relating to distributed TOEs are specified for this TOE.

FAU_GEN.1 Audit data generation

FAU_GEN.1 Test #1

FAU_GEN.1

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.

- Test Steps:

    - NO STEPS: No separate testing needed as all necessary log generation is exercised in other test activities.

    - The TOE is not a distributed TOE

FAU_GEN.2 User identity association

FAU_GEN.2 Test #1

FAU_GEN.2

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

- N/A – Tested in conjunction with FAU_GEN.1 above.

FAU_ STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1

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

FAU_STG_EXT.1 Test #1.1a

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.

- Test Steps:
    - TOE: Save copy OF configuration.
    - Linux Server: Configure test server
    - Linux Server: Start Syslog server/service.
    - Linux Server: Start collection of PCAP for UDP/514
    - TOE: Configure ACOS for Syslog (see above)
    - Observe: Traffic on PCAP on DUT shell (should be encrypted)
    - Observe: Traffic on PCAP @ Linux Server
    - TOE: generate event log by logging in/logging out.
    - Observe: Verify login/logout events are received by syslog server
    - Record: PCAP of exchanges from TOE and Linux Server
    - ACOS Record: CLI outputs and ACOS Logged Events

- TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated transfer to and receipt of audit data by external server
  - Demonstrated Syslog communications to the target server for ACOS audit and event records
  - IPSec encryption & configuration per AGD of this communication channel further demonstrated in FTC_ITC.1 Test #3
- This satisfies the testing requirement.

FAU_STG_EXT.1 Test #1.2a

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).

- Test Steps:
  - TOE: Save copy OF configuration.
  - TOE Configure ACOS for audit and event log
  - TOE: Generate event logs until buffer is full and demonstrate log overwrite oldest record 1st when full.
  - TOE: Generate audit logs until buffer is full and demonstrate log rotation oldest record 1st when full.
  - ACOS Record: CLI outputs and ACOS Logged Events
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated TOE behavior when local audit log is full
  - Demonstrated that oldest records are overwritten for new audit records when audit log space is exceeded.
  - Demonstrated TOE behavior when event log is full

    - Demonstrated that oldest records are overwritten for new event records when event log space is exceeded.
- This satisfies the testing requirement.

FAU_STG_EXT.1 Test #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~~

    - NOTE:
        1. FAU_STG_EXT.2/LocSpace not claimed

~~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.~~

- The TOE is not a distributed TOE

4.2.2 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation (Refinement)

FCS_CKM.1 Test #1a - Syslog

- Test Steps:
    - TOE: Save copy OF configuration.

    - Linux Server: Configure test server (as described above)
    - Linux Server: Start Syslog server/service.
    - Linux Server: Start collection of PCAP for UDP/514-for packets to/from DUT

    - TOE: Configure ACOS for Syslog (see above)
    - TOE: Login/logout of the system to generate syslog
    - Observe: Traffic on PCAP @ Linux Server
    - Observe: Syslog received on server

    - TOE: Unconfigure syslog server
    - TOE: Generate logs on ACOS
    - Observe: There should be no logs received by server.

    - Record: PCAP of exchanges from TOE to Linux Server

- ACOS Record: CLI outputs and ACOS Logged Events

- Result:
- **Verdict: Pass**
  - Demonstrated Diffie-Hellman Group 14 and FFC Schemes using a “safe-prime” group for IPsec tunnel
  - Demonstrated RSA-based key establishment

  - Demonstrated interoperability with a known, good, non-TOE IPsec implementation supporting DH group 14 and where IPsec is the only claimed protocol for FTP_TRP.1/Admin and FTP_ITC

  - Demonstrated tunnel mode interoperability with an IPsec implementation integrated on a server in the operational environment.

  - Demonstrated Syslog trusted channel communication for remote syslog.

- This satisfies the testing requirement.

FCS_CKM.1 Test #1b – NTP

- Test Steps:
  - Linux Server: Configure test server (as described above)
  - Linux Server: Start NTP server/service.
  - Linux Server: Start collection of PCAP for UDP/123

  - TOE: show ACOS system time
  - Observe: ACOS system time is from local device clock settings
  - TOE: Configure ACOS for NTP (see above)
  - TOE: “enable” synchronization with NTP Server #1
  - Observe: Traffic on PCAP @ Linux Server
  - Observe: NTP Status (`show ntp status`) for the NTP Servers
  - Observe: NTP Servers #1 indicates “synchronized” status.
  - Observe: ACOS system time is consistent with the NTP Server #1 network time
  - TOE: Remove NTP configuration from ACOS
  - Observe: Verify ACOS is not communicating with 10.1.1.167 for NTP
  - Record: PCAP of exchanges from TOE and Linux Server
  - ACOS Record: CLI outputs and ACOS Logged Events

- Result:
- **Verdict: Pass**
  - Continuation of test FCS_CKM.1 Test #1a

        1. Demonstrated interoperability with a known, good, non-TOE IPsec implementation supporting DH group 14 and where IPsec is the only claimed protocol for FTP_TRP.1/Admin and FTP_ITC

        2. Demonstrated tunnel mode interoperability with an IPsec implementation integrated on a server in the operational environment.

    - Demonstrated NTP trusted channel communication for remote NTP server.

- This satisfies the testing requirement.

FCS_CKM.1 Test #1c – SCP

- Test Steps:

    - Linux Server: Configure test server (as described above)
    - Linux Server: Set up and start (if needed) SCP service.
    - Linux Server: Start collection of PCAP for TCP/22
    - TOE: Import something innocuous (link an AFLEX file) with SCP
    - TOE: Export something innocuous (link an AFLEX file) with SCP
    - Observe: Traffic on PCAP @ Linux Server
    - Observe: File transfer operations SUCCEEDS
    - TOE: REPEAT IMPORT/EXPORT w/ SCP ABOVE with bad remote filename
    - Observe: Both import operations FAIL (including logged reason for failure)
    - TOE: REPEAT IMPORT/EXPORT w/ SCP ABOVE with bad password
    - Observe: Both import operations FAIL (including logged reason for failure)
    - TOE: REPEAT IMPORT/EXPORT w/ SCP ABOVE with bad user name
    - Observe: Both import operations FAIL (including logged reason for failure)
    - TOE: REPEAT IMPORT/EXPORT w/ SCP ABOVE with bad IP address of Server
    - Observe: Both import operations FAIL (including logged reason for failure)
    - Record: PCAP of exchanges from TOE and Linux Server
    - ACOS Record: CLI outputs and ACOS Logged Events
    - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - Continuation of test FCS_CKM.1 Test #1a
        1. Demonstrated interoperability with a known, good, non-TOE IPsec implementation supporting DH group 14 and where IPsec is the only claimed protocol for FTP_TRP.1/Admin and FTP_ITC
        2. Demonstrated tunnel mode interoperability with an IPsec implementation integrated on a server in the operational environment.

  - Demonstrated SCP/SFTP file server trusted channel communication for file transfer servers.

- This satisfies the testing requirement.

FCS_CKM.1

FCS_CKM.1 Test #1d

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.

Diffie-Hellman Group 14 and FFC Schemes using “safe-prime” groups

Testing for FFC Schemes using Diffie-Hellman group 14 and/or safe-prime groups is done as part of testing in CKM.2.1.

FFC Schemes using “safe-prime” groups

TD0580 – Testing for FFC Schemes using safe-prime groups is done as part of testing in CKM.2.1.

- **Verdict: Pass**

RSA key generation is covered by the following CAVP certificates: C1940 and A1305. ECDSA KeyGen and KeyVer are covered by the following CAVP certificates: C1198 and C1940.

FCS_CKM.2 Cryptographic Key Establishment (Refinement)

FCS_CKM.2 Test #1

- N/A – Tested in conjunction with test cases performed as indicated under FCS_CKM.1 (Tests #1a, 1b, 1c) to demonstrate interoperability with a known, good implementation supporting Diffie-Hellman Group 14.

(TD0580 applied)

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 schemes

The evaluator shall verify the correctness of the TSF's implementation of RSAESPKCS1-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.

FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4 Test #1

~~None~~

- N/A – No test requirement specified in NDcPPv2.2e SD.

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

FCS_COP.1/DataEncryption

FCS_COP.1/DataEncryption Cryptographic Operation Test #1

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 AESCBC 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 AESCBC-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 nonzero 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. Since the Counter Mode does not specify the counter that is used, it is not possible to implement an automated test for this mode. The generation and management of the counter is tested through FCS_SSH*_EXT.1.4. If CBC and/or GCM are selected in

FCS_COP.1/DataEncryption, the test activities for those modes sufficiently demonstrate the correctness of the AES algorithm. If CTR is the only selection in FCS_COP.1/DataEncryption, the AES-CBC Known Answer Test, AESGCM 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.

- **Verdict: Pass**

AES is covered by the following CAVP certificates: C1198, C1940 and A1181.

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

FCS_COP.1/SigGen

FCS_COP.1/ SigGen Cryptographic Operation Test #1

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.

- **Verdict: Pass**

RSA SigGen is covered by the following CAVP certificates: C1198 and C1940. RSA SigVer is covered by the following CAVP certificate: A1305. ECDSA SigGen and SigVer is covered by the following CAVP certificates: C1198 and C1940.

FCS_COP.1/ Hash Cryptographic Operation (Hash Algorithm)

FCS_COP.1/Hash

FCS_COP.1/Hash Cryptographic Operation Test #1

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.

- **Verdict: Pass**

SHS is covered by the following CAVP certificates: C1198, C1940 and A1181.

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

A.2.2.6 FCS_COP.1/KeyedHash

FCS_COP.1/ KeyedHash Cryptographic Operation Test #1

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.

- **Verdict: Pass**

HMAC is covered by the following CAVP certificates: C1198, C1940 and A1181.

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

FCS_RBG_EXT.1

FCS_ RBG_EXT.1 Extended: Cryptographic Operation Test #1

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

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.

- **Verdict: Pass**

DRBG-CTR is covered by the following CAVP certificates: C1198 and C1940.

FCS_IPSEC_EXT.1 IPsec Protocol

FCS_IPSEC_EXT.1 Test Common

A.2.2.10 FCS_IPSEC_EXT.1

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

FCS_IPSEC_EXT.1.1 Test #1

FCS_IPSEC_EXT.1.1

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.

- Test Steps:
  - TOE: Save copy OF configuration.
  - IPSec Test Peer: Configure/set-up the IPSec tunnel, including verifiable certificate

- TOE: Configure/set-up the IPSec tunnel, including verifiable certificate and with certificate verification using OCSP or CRL.
- Observe: ACOS validated the IPSec test peer's certificate and that the IPSec tunnel is "up"
- TOE: Enable ACOS debug packet, tunnel, and monitor (as necessary to confirm packets dropped and passed through the tunnel)
- TOE: Add ACL - deny all ICMP traffic from IPsec protected subnetwork
- Observe: Cannot ping **from** IPsec protected subnetwork (packets from IPsec subnet)
- Record: TOE settings, CLI outputs, and ACOS Logged Events
- TOE: Add ACL – permit ICMP type 8 (echo request) from the IPsec protected subnetwork
- Observe: Pings works to and from IPsec protected subnetwork
- Observe: Cannot ping to IPsec protected subnetwork
  (ICMP echo requests pass, but responses are dropped)
- Record: TOE settings, CLI outputs, and ACOS Logged Events
- TOE: Add ACL – permit ICMP type 0 (echo reply) from the IPsec protected subnetwork
- Observe: Pings works to and from IPsec protected subnetwork
- Record: TOE settings, CLI outputs, and ACOS Logged Event
- TOE: Add ACLs – deny all SSH traffic (TCP/22) from IPsec protected subnetwork
- Observe: Can't SSH to or from device (IP address) on IPsec protected subnetwork
- Record: TOE settings, CLI outputs, and ACOS Logged Event

- TOE: Add ACL – permit SSH traffic for 1 server (IP address) on IPsec protected subnetwork
- Observe: SSH works to and from (IP address) on IPsec protected subnetwork, but not from others addresses.
- Record: TOE settings, CLI outputs, and ACOS Logged Event
- TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated ability to affect SPD PERMIT/DENY using ACL with IP Addresses, protocol types, and ports through the IPsec tunnel
  - Demonstrated ability to determine/audit packet operation and dispositions through packet debugging.
  - BYPASS using ACL with IP Addresses, protocol types, and ports for clear text connections is tested in FTP_TRP.1/Admin Test #2b
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.1 Test #2

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.
  1. 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.

- Test Steps:
  - TOE: Save copy OF configuration.
  - TOE: Restore final configuration recorded from FCS_IPSEC_EXT.1.1 Test #1
  - TOE: Add rules – deny SSH traffic for 1 server (IP address) on IPsec protected subnetwork
  - Observe: Can't SSH to or from device (IP address) on IPsec protected subnetwork
  - Record: TOE settings, CLI outputs, and ACOS Logged Event

  - TOE: Add rules – permit SSH traffic for 1 server (IP address) on IPsec protected subnetwork
  - Observe: SSH works to and from (IP address) on IPsec protected subnetwork, but not from others addresses
  - Record: TOE settings, CLI outputs, and ACOS Logged Event

  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated ACL (SPD rule) precedence where overlapping rules
  - Demonstrated ability to determine/audit packet operation and dispositions through packet debugging.
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.2 Test #1

FCS_IPSEC_EXT.1.2

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

- N/A – Tested in conjunction with FCS_IPSEC_EXT.1.1 above, FTP_ITC.1 Test 3a/3b, and FTP_TRP.1/Admin Test 2a tests where default discard rules were applied.

FCS_IPSEC_EXT.1.3 Test #1

FCS_IPSEC_EXT.1.3

The evaluator shall **perform the following test(s)** based on the selections chosen:

a) Test 1: If tunnel mode is selected, the evaluator uses the guidance documentation to configure the TOE to operate in tunnel mode and also configures a VPN peer to operate in tunnel mode.

- The evaluator configures the TOE and the VPN peer to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated.
- The evaluator shall then initiate a connection from the TOE to connect to the VPN peer.
- The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the tunnel mode.

- Test Steps:
  - TOE: Enable ike-logging-enable configuration option
  - TOE: “disable” tunnel to IPsec Test Peer
  - IPsec Test Peer: confirm configured ONLY for ‘tunnel mode’
  - TOE: ‘enable” tunnel to IPsec Test Peer
  - Observe: tunnel establishment **SUCCEEDS**
  - Record: TOE and IPsec Test Peer config settings
  - Record: PCAP of the test’s packet exchanges w/ IPsec Test Peer.
  - Record: CLI outputs and ACOS Logged Events
  - IPsec Test Peer: Restore configuration to support ONLY ‘tunnel mode”

- Result:
- **Verdict: Pass**
  - Demonstrated tunnel mode IPsec session with VPN peer
  - Demonstrated IPsec tunnel stablishment initiated by the TOE
  - Demonstrated IPsec tunnel termination initiated by the TOE
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.3 Test #2

~~**b) Test 2: If transport mode is selected**, the evaluator uses the guidance documentation to configure the TOE to operate in transport mode and also configures a VPN peer to operate in transport mode. The evaluator configures the TOE and the VPN peer to use any of the allowed cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator then initiates a connection from the TOE to connect to the VPN peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the transport mode.~~

- NOTE:
    1. Transport mode not claimed

FCS_IPSEC_EXT.1.4 Test #1

FCS_IPSEC_EXT.1.4

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.

- Algorithms List :==
    - encryption aes-256 hash sha512
    - encryption aes-192 hash sha384
    - encryption aes-128 hash sha256
    - encryption aes-128 hash sha1
    - encryption aes-gcm-256 hash sha512
    - encryption aes- gcm-192 hash sha384
    - encryption aes- gcm-128 hash sha256

- Test Steps:
    - IPsec Test Peer: Save copy OF configuration.
    - TOE: Save copy OF configuration.

    - For EACH OF THE Supported IKE Versions (IKEv2)
        1. For EACH OF THE settings in the Algorithms List
            - TOE: "disable" tunnel to IPsec Test Peer
            - IPsec Test Peer: Modify configuration to support encryption/hashing for the setting.
            - TOE: Modify "vpn ike-gateway" configuration for ONLY the encryption and hash.
            - TOE: Modify "vpn ipsec" configuration for ONLY the encryption and hash.
            - TOE: "enable" tunnel to IPsec Test Peer
            - Observe: tunnel establishment **SUCCEEDS**
            - Record: TOE and IPsec Test Peer config settings
            - Record: PCAP of the test's packet exchanges w/ IPsec Test Peer.
            - Record: CLI outputs and ACOS Logged Events

    - IPsec Test Peer: Restore configuration to saved settings.
    - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - Demonstrated ability to establish IPsec tunnels for the claimed encryption and HMAC algorithms claimed.

- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.5 Test #1a

FCS_IPSEC_EXT.1.5

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.~~

- N/A – IKEv1 has been removed from the claimed scope

FCS_IPSEC_EXT.1.5 Test #1b

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 Steps:

TOE:	"disable" tunnel to IPsec Test Peer
TOE:	"enable" tunnel to IPsec Test Peer
Observe:	tunnel establishment **SUCCEEDS**
TOE:	establish SSH session on Test Subnet D
Observe:	IPsec NAT'ing used
Record:	IPsec Test Peer config settings
Record:	PCAPs from TOE and SSH Server (to confirm NAT)
Record:	CLI outputs and ACOS Logged Events
Routers:	Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated ability to establish IPsec tunnel using IKEv2
  - Demonstrated ability to establish IPsec tunnel using IKEv2 with NAT Traversal with encapsulation of the tunnel under UDP/4500.
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.6 Test #1

FCS_IPSEC_EXT.1.6

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 Steps:
  - NO STEPS: Test uses results from FCS_IPSEC_EXT.1.4 Test #1

FCS_IPSEC_EXT.1.7 Test #1a

FCS_IPSEC_EXT.1.7

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.~~

- N/A – IKEv1 Phase 1 SA & IKEv2 SA lifetimes limitations by content/volume (bytes) is not claimed.

FCS_IPSEC_EXT.1.7 Test #1b

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 of the TOE. The evaluator shall establish a SA between the TOE and the test peer, maintain the Phase 1 SA for 24 hours, and determine that a new Phase 1 SA is negotiated on or before 24 hours has elapsed.

- The evaluator shall verify that the TOE initiates a Phase 1 negotiation.

(TD0634 applied)

- Test Steps:
  - TOE: "disable" tunnel to IPsec Test Peer
  - IPsec Test Peer: Custom modification A10-IPsec-VPN #1 (for IPsec Test Peer) to "*Disable IPsec lifetime to based on time*" before renegotiating new IKE SA and child ESP SA"

- IPsec Test Peer: Load and restart with ‘MODIFIED ACOS VPN #1 including custom modified IPsec lifetime.
- TOE: “enable” tunnel to IPsec Test Peer
- TOE: … DO NOTHING for 24 hours …

- Observe: @ 24 hours, the Phase 1 SA is negotiated
- Observe: Negotiation is initiated by the TOE
- Record: CLI outputs and ACOS Logged Events
- IPsec Test Peer: Restore Standard A10-IPsec-VPN (for IPsec Test Peer) Implementation

- Result:
- **Verdict: Pass**
  - Demonstrated that prior to 24-hours the TOE initiates re-keying of the Phase 1 SA
  - Demonstrated that re-keying of the Phase 1 SA is successful prior to 24 hours.
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.8 Test #1a

FCS_IPSEC_EXT.1.8

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 2 negotiation.

- Test Steps:
  - TOE: “disable” tunnel to IPsec Test Peer
  - IPsec Test Peer: Save copy OF configuration.
  - TOE: Save copy OF configuration.
  - IPsec Test Peer: Modify configuration to support equivalent of ‘lifebytes” to be > 10 GBytes
  - TOE: Modify configuration to for ‘lifebytes’ to be 10 GBytes
  - TOE: “enable” tunnel to IPsec Test Peer
  - TOE: Generate over > 10 GB e IPSec Tunnel

- ○ Observe: @ or prior to 10 GB, the SA is negotiated
- ○ Observe: Negotiation is initiated by the TOE
- ○ Record: CLI outputs and ACOS Logged Events
- ○ TOE: Restore configuration to saved settings.
- ○ IPsec Test Peer: Restore Standard A10-IPsec-VPN (for IPsec Test Peer) Implementation

- Result:
- **Verdict: Pass**
  - Demonstrated that prior to the configured lifebytes (lifetime in bytes) claimed and included in the AGD that the TOE initiates re-keying of the Phase 2 SA @ before 10 GB maximum.
  - Demonstrated that re-keying of the Phase 2 SA is successful prior to maximum configured data volume occurs.
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.8 Test #1b

b) Test 2: If 'length of time' is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime no later than 8 hours for the Phase 2 SA following the guidance documentation. The evaluator shall configure a test peer with a Phase 2 SA lifetime that exceeds the Phase 2 SA lifetime of the TOE. The evaluator shall establish a SA between the TOE and the test peer, maintain the Phase 1 SA for 8 hours, and determine that once a new Phase 2 SA is negotiated when or before 8 hours has lapsed.

- The evaluator shall verify that the TOE initiates a Phase 2 negotiation.

(TD0634 applied)

- Test Steps:
  - ○ TOE: "disable" tunnel to IPsec Test Peer
  - ○ IPsec Test Peer: Custom modification A10-IPsec-VPN #2 (for IPsec Test Peer) to "*Disable IPsec lifetime to based on time"* before renegotiating new IKE Phase 2 SA and IKEv2 Child SA"
  - ○ IPsec Test Peer: Load and restart with 'MODIFIED ACOS VPN #2 including custom modified IPsec lifetime.
  - ○ TOE: "enable" tunnel to IPsec Test Peer
  - ○ TOE: … DO NOTHING for 8 hours …
  - ○ Observe: @ 8 hours, the Phase 2 SA is negotiated
  - ○ Observe: Negotiation is initiated by the TOE
  - ○ Record: CLI outputs and ACOS Logged Events
  - ○ IPsec Test Peer: Restore Standard A10-IPsec-VPN (for IPsec Test Peer) Implementation

- Result:
- **Verdict: Pass**
  - Demonstrated that prior to 8-hours the TOE initiates re-keying of the Phase 2 SA
  - Demonstrated that re-keying of the Phase 2 SA is successful prior to 8 hours.
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.9 Test #1

-

- Procedure:
  - NO STEPS: No tests are specified for FCS_IPSEC_EXT.1.9

FCS_IPSEC_EXT.1.10 Test #1a

FCS_IPSEC_EXT.1.10

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.

- N/A – the first selection is not chosen for this SFR.

FCS_IPSEC_EXT.1.10 Test #1b

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.

- Test Steps:
  - NO STEPS: No operational testing involved, as test specification is verification of TSS in the ST. The following "Observations" is provided to comply with this test statement.

- Result:
- **Verdict: Pass**
  - Section 7.1.2.9 of the ST states that the TOE uses AES-CTR DRBG to generate the secret value 'x' used in the IKEv2 Diffie-Hellman key exchange ("x" in g^x mod p). These exponents generated for DH Group 14 are 224-bits (28 bytes) long. Nonces generated using AES-CTR DRBG are 256-bits (32-bytes), which is more than 128-bits in size and half of the largest output size supported (SHA-512).
  - As such, PRF generated nonces satisfy the requirement since they are more than 128 bits in length AND half the largest output size supported (SHA-512).

- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.11 Test #1

FCS_IPSEC_EXT.1.11

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.

- DH Group List :==
  - dh-group 14

- Test Steps:
  - IPsec Test Peer: Save copy OF configuration.
  - TOE: Save copy OF configuration.
  - Modify configuration to support **IKEv2** @ both IPsec Test Peer & TOE
  - For EACH OF THE settings in the DH Group List
    1. TOE: “disable” tunnel to IPsec Test Peer
    2. IPsec Test Peer: Modify configuration to DH Group for the setting.
    3. TOE: Modify “vpn ike-gateway” configuration for ONLY the DH Group.
    4. TOE: Modify “vpn ipsec” configuration for ONLY the DH Group .
    5. TOE: “enable” tunnel to IPsec Test Peer
    6. Observe: tunnel establishment SUCCEEDS
    7. Record: TOE and IPsec Test Peer config settings
    8. Record: PCAP of the test’s packet exchanges w/ IPsec Test Peer.
    9. Record: CLI outputs and ACOS Logged Events

  - IPsec Test Peer: Restore Standard A10-IPsec-VPN (for IPsec Test Peer) Implementation
  - TOE: Restore original/reference settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that DH Group 14 claimed with IKE v2 as restricted by guidance
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.12 Test #1

FCS_IPSEC_EXT.1.12

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.

- Test Steps:
  - NO STEPS: Test uses results from FCS_IPSEC_EXT.1.4 Test #1

FCS_IPSEC_EXT.1.12 Test #2

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.

- Algorithms Scenarios List :==

vpn ike-gateway Algorithm	**vpn ipsec Algorithm**	
encryption aes-192 hash sha384	encryption aes-192 hash sha512	(ipsec hash stronger)
encryption aes-128 hash sha384	encryption aes-192 hash sha384	(ipsec enc stronger)

- Test Steps:
  - IPsec Test Peer: Save copy OF configuration.
  - TOE: Save copy OF configuration.
  - TOE: Enable ike-logging-enable configuration option
  - TOE: Configure “vpn ipsec-cipher-check”.
  - For EACH IKE VERSION supported (IKEv2)
    1. For EACH OF THE settings in the Algorithms Scenarios List
       - TOE: “disable” tunnel to IPsec Test Peer
       - IPsec Test Peer: Modify configuration to support encryption/hashing for the setting.
       - TOE: Modify “vpn ike-gateway” config for the indicated encryption and hash.
       - TOE: Modify “vpn ipsec” config for the indicated encryption and hash.
       - TOE: “enable” tunnel to IPsec Test Peer
       - Observe: tunnel establishment **FAILS**
       - Record: TOE and IPsec Test Peer config settings
       - Record: CLI outputs and ACOS Logged Events
  - IPsec Test Peer: Restore configuration to saved settings.
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**

  - Demonstrated that when IPsec tunnel is configured tunnel stronger/longer key sizes (either for SHA hash or AES encryption) in the IPsec GW configuration than the IKE GW configuration, that the IPsec tunnel fails to be instantiated if the negotiated algorithms result in condition where the outer (ESP) SA's key size exceeds that of the inner (IKE) SA's key size.

- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.12 Test #3

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.

- Test Steps:
  - IPsec Test Peer: Save copy OF configuration.
  - TOE: Enable ike-logging-enable configuration option
  - TOE: Configure "vpn ipsec-cipher-check".
  - TOE: "disable" tunnel to IPsec Test Peer
  - IPsec Test Peer: Modify **"vpn ike-gateway"** config to support "encryption 3des hash sha1.
  - TOE: "enable" tunnel to IPsec Test Peer
  - Observe: tunnel establishment **FAILS**
  - Record: TOE and IPsec Test Peer config settings
  - TOE: "disable" tunnel to IPsec Test Peer
  - IPsec Test Peer: Modify **"vpn ike-gateway"** config to support "encryption aes-128 hash md5.
  - TOE: "enable" tunnel to IPsec Test Peer
  - Observe: tunnel establishment **FAILS**
  - Record: TOE and IPsec Test Peer config settings
  - Record: CLI outputs and ACOS Logged Events
  - IPsec Test Peer: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that when VPN peer does not support claimed IKE (aka IKE GW) algorithms for encryption and hash the IPsec tunnel fails to be instantiated.

- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.12 Test #4

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 Steps:
  - IPsec Test Peer: Save copy OF configuration.
  - TOE: Enable ike-logging-enable configuration option
  - TOE: Configure “vpn ipsec-cipher-check”.
  - TOE: “disable” tunnel to IPsec Test Peer
  - IPsec Test Peer: Modify **“vpn ipsec”** config to support “encryption 3des hash sha1.
  - TOE: “enable” tunnel to IPsec Test Peer
  - Observe: tunnel establishment **FAILS**
  - Record: TOE and IPsec Test Peer config settings
  - TOE: “disable” tunnel to IPsec Test Peer
  - IPsec Test Peer: Modify **“vpn ipsec”** config to support “encryption aes-128 hash md5.
  - TOE: “enable” tunnel to IPsec Test Peer
  - Observe: tunnel establishment **FAILS**
  - Record: TOE and IPsec Test Peer config settings
  - Record: CLI outputs and ACOS Logged Events
  - IPsec Test Peer: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that when VPN peer does not support claimed ESP (aka IPsec GW) algorithms for encryption and hash the IPsec tunnel fails to be instantiated.
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.13 Test #1

FCS_IPSEC_EXT.1.13

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.

- N/A – Testing performed in FIA_X509_EXT.1, FIA_X509_EXT.2 (for IPsec connections), and FCS_IPSEC_EXT.1.1.

FCS_IPSEC_EXT.1.14 Test #1

FCS_IPSEC_EXT.1.14

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.~~

- N/A – The TOE does not claim CN's.

FCS_IPSEC_EXT.1.14 Test #2a

• **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 DNSID, CN with incorrect DNS-ID (and not a different type of identifier)) and verify that IKE authentication succeeds.~~

- Test Steps:
  - IPsec Test Peer: Save copy OF configuration.
  - TOE: Save copy OF configuration.
  - TOE: "disable" tunnel to IPsec Test Peer
  - IPsec Test Peer: Configure for RSA Authentication and certificate with **SAN = FQDN of the IPsec test peer** and **DN = DN of the IPsec test peer**.
    1. **local-id :== FQDN of the IPSec test peer**
    2. remote-id :== FQDN of TOE in the TOE RSA certificate
  - TOE: Configure "vpn ike-gateway" for test peer with
    1. auth-method :== rsa-signature
    2. **remote-id :== FQDN of the IPsec test peer** (shouldn't match Cert's DN, should match SAN)
    3. remote-ip :== IPv4 address of IPsec test peer
    4. local-id :== FQDN of TOE in the local RSA certificate
    5. local-ip :== IPv4 address of TOE
    6. key / local-cert : == test_dut _rsa_1.key/.crt(from signed RSA cert/key import)
  - TOE: "enable" tunnel to IPsec Test Peer
  - Observe: tunnel establishment **SUCCEEDS**
  - Record: TOE and IPsec Test Peer config settings
  - Record: CLI outputs and ACOS Logged Events
  - IPsec Test Peer: Restore configuration to saved settings.
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**

    - Demonstrated that for reference IDs are configured for SAN = FQDN on VPN test peer that the IPsec tunnel (in particular, the IKE authentication) succeeds.
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.14 Test #2b

• **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 DNSID, CN with incorrect DNS-ID (and not a different type of identifier)) and verify that IKE authentication succeeds.~~

- Test Steps:
    - IPsec Test Peer: Save copy OF configuration.
    - TOE: Save copy OF configuration.
    - TOE: "disable" tunnel to IPsec Test Peer
    - IPsec Test Peer: Configure for RSA Authentication and certificate with **SAN = IP Address of the IPsec test peer** and **DN = DN of the IPsec test peer**.
        1. **local-id :== IP Address of the IPSec test peer**
        2. remote-id :== FQDN of TOE in the TOE RSA certificate
    - TOE: Configure "vpn ike-gateway" for test peer with
        1. auth-method :== rsa-signature
        2. **remote-id :== IP Address of the IPsec test peer** (shouldn't match Cert's DN, should match SAN)
        3. remote-ip :== IPv4 address of IPsec test peer
        4. local-id :== FQDN of TOE in the local RSA certificate
        5. local-ip :== IPv4 address of TOE
        6. key / local-cert : == test_dut _rsa_1.key/.crt(from signed RSA cert/key import)
    - TOE: "enable" tunnel to IPsec Test Peer
    - Observe: tunnel establishment **SUCCEEDS**
    - Record: TOE and IPsec Test Peer config settings
    - Record: CLI outputs and ACOS Logged Events
    - IPsec Test Peer: Restore configuration to saved settings.
    - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - Demonstrated that for reference IDs are configured for SAN = IP-ADDRESS on VPN test peer that the IPsec tunnel (in particular, the IKE authentication) succeeds.

- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.14 Test #3

• ~~**Test 3:** [conditional] **For each CN/identifier type combination selected**, the **evaluator shall:**~~

~~e) Create a valid certificate with the CN so it contains the valid identifier followed by '\0'.~~

~~- **If the TOE prioritizes CN checking over SAN** (through explicit specification of the field when specifying the reference identifier or prioritization rules) for the same identifier type, *the evaluator shall configure the SAN so it matches the reference identifier*.~~

~~f) Configure the peer's reference identifier on the TOE (per the administrative guidance) to match the CN without the '\0' and~~

~~- verify that IKE authentication fails.~~

- N/A – The TOE does not claim CN's.

FCS_IPSEC_EXT.1.14 Test #4a

• **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 Steps:
  - IPsec Test Peer: Save copy OF configuration.
  - TOE: Save copy OF configuration.
  - TOE: Enable ike-logging-enable configuration option
  - TOE: "disable" tunnel to IPsec Test Peer
  - IPsec Test Peer: Configure for RSA Authentication and certificate with **SAN = FQDN of the IPsec test peer** and **DN = DN of the IPsec test peer**.
    1. **local-id :== DN of the peer**
    2. remote-id :== FQDN of TOE in the TOE RSA certificate

- TOE: Configure “vpn ike-gateway” for test peer with
    1. auth-method :== rsa-signature
    2. **remote-id** **:== different DN than on the peer** (shouldn’t match Cert’s DN, shouldn’t match SAN)
    3. remote-ip :== IPv4 address of IPsec test peer
    4. local-id :== FQDN of TOE in the local RSA certificate
    5. local-ip :== IPv4 address of TOE
    6. key / local-cert : == test_dut _rsa_1.key/.crt(from signed RSA cert/key import)
- TOE: “enable” tunnel to IPsec Test Peer
- Observe: tunnel establishment **FAILS**
- Record: TOE and IPsec Test Peer config settings
- Record: CLI outputs and ACOS Logged Events
- IPsec Test Peer: Restore configuration to saved settings.
- TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - Demonstrated that for VPN test peer with identifier in the SAN of its certificate which does not match the remote identifier configured on the DUT for the peer that the IPsec tunnel (in particular, the IKE authentication) **FAILS**.
        1. Showed that the ACOS detects mismatch between the SAN present in the certificate and the configured remote-id for the peer for **differing FQDN** value.
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.14 Test #4b

• **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 Steps:
    - IPsec Test Peer: Save copy OF configuration.
    - TOE: Save copy OF configuration.

- ○ TOE: Enable ike-logging-enable configuration option
- ○ TOE: “disable” tunnel to IPsec Test Peer
- ○ IPsec Test Peer: Configure for RSA Authentication and certificate with **SAN = IP Address of the IPsec test peer** and **DN = IP Address of the IPsec test peer**.
    1. **local-id :== IP address of the peer**
    2. remote-id :== FQDN of TOE in the TOE RSA certificate
- ○ TOE: Configure “vpn ike-gateway” for test peer with
    1. auth-method :== rsa-signature
    2. **remote-id :== different IP address than on the peer** (shouldn’t match Cert’s DN, shouldn’t match SAN)
    3. remote-ip :== IPv4 address of IPsec test peer
    4. local-id :== FQDN of TOE in the local RSA certificate
    5. local-ip :== IPv4 address of TOE
    6. key / local-cert : == test_dut _rsa_1.key/.crt(from signed RSA cert/key import)
- ○ TOE: “enable” tunnel to IPsec Test Peer
- ○ Observe: tunnel establishment **FAILS**
- ○ Record: TOE and IPsec Test Peer config settings
- ○ Record: CLI outputs and ACOS Logged Events
- ○ IPsec Test Peer: Restore configuration to saved settings.
- ○ TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - ○ Demonstrated that for VPN test peer with identifier in the SAN of its certificate which does not match the remote identifier configured on the DUT for the peer that the IPsec tunnel (in particular, the IKE authentication) **FAILS**.
        1. Showed that the ACOS detects mismatch between the SAN present in the certificate and the configured remote-id for the peer for differing **IP address** value.
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.14 Test #5

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 Steps:
    - ○ IPsec Test Peer: Save copy OF configuration.
    - ○ TOE: Save copy OF configuration.
    - ○ TOE: “disable” tunnel to IPsec Test Peer
    - ○ IPsec Test Peer: Configure for RSA Authentication and certificate with **NO SAN** and **DN = DN of the IPsec test peer**.
        1. **local-id :== DN of the IPSec test peer**

        2. **remote-id :== FQDN of TOE in the TOE RSA certificate**

    - TOE: Configure “vpn ike-gateway” for test peer with
        1. auth-method :== rsa-signature
        2. **remote-id :== DN of the IPsec test peer** (should match Cert’s DN, cannot match SAN)
        3. remote-ip :== IPv4 address of IPsec test peer
        4. **local-id :== FQDN of TOE in the local RSA certificate**
        5. local-ip :== IPv4 address of TOE
        6. key / local-cert : == test_dut _rsa_1.key/.crt(from signed RSA cert/key import)

    - TOE: “enable” tunnel to IPsec Test Peer
    - Observe: tunnel establishment **SUCCEEDS**
    - Record: TOE and IPsec Test Peer config settings
    - Record: CLI outputs and ACOS Logged Events
    - IPsec Test Peer: Restore configuration to saved settings.
    - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - Demonstrated that for reference IDs are configured for SAN = FQDN on VPN test peer matching DN in the peer’s certificate that the IPsec tunnel (in particular, the IKE authentication) SUCCEEDS.
        1. DUT will be trying to match the configured remote ID with the subject of the peer certificate.
        2. Whether the peer certificate has or does not have a SAN, does not matter for this test.
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.14 Test #6a

• **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.

- Test Steps:
    - IPsec Test Peer: Save copy OF configuration.
    - TOE: Save copy OF configuration.
    - TOE: Enable ike-logging-enable configuration option
    - TOE: “disable” tunnel to IPsec Test Peer
    - IPsec Test Peer: Configure for RSA Authentication and certificate with **NO SAN** and **DN = DN of the IPsec test peer** with TWO CNs
        1. **local-id :== DN of the IPSec test peer w/ only ONE CN**
        2. **remote-id :== FQDN of TOE in the TOE RSA certificate**

- TOE: Configure "vpn ike-gateway" for test peer with
    1. auth-method :== rsa-signature
    2. **remote-id** **:== DN of the IPsec test peer w/ only ONE CN** (shouldn't match Cert's DN, can't match SAN)
    3. remote-ip :== IPv4 address of IPsec test peer
    4. **local-id** **:== FQDN of TOE in the local RSA certificate**
    5. local-ip :== IPv4 address of TOE
    6. key / local-cert : == test_dut _rsa_1.key/.crt(from signed RSA cert/key import)

- TOE: "enable" tunnel to IPsec Test Peer
- Observe: tunnel establishment **FAILS**
- Record: TOE and IPsec Test Peer config settings

- Record: CLI outputs and ACOS Logged Events
- IPsec Test Peer: Restore configuration to saved settings.
- TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - Demonstrated that for VPN test peer with FQDN containing two copies of the CN matching DN in the peer's certificate that the IPsec tunnel (in particular, the IKE authentication) **FAILS**.
        1. Showed that comparison is on the entire CN value and not just the first part value which would normally match if not for 2nd copy of the CN.
- This satisfies the testing requirement.

FCS_IPSEC_EXT.1.14 Test #6b

b) Append '\0' to a non-CN field of an otherwise authorized DN.

- Test Steps:
    - IPsec Test Peer: Save copy OF configuration.
    - TOE: Save copy OF configuration.
    - TOE: Enable ike-logging-enable configuration option
    - TOE: "disable" tunnel to IPsec Test Peer
    - IPsec Test Peer: Configure for RSA Authentication and certificate with **NO SAN** and **DN = DN of the peer** and NULL ('\0') terminated**)**
        1. **local-id** **:== DN of the IPSec test peer** (excluding NULL ("\0") terminator
        2. remote-id :== FQDN of TOE in the TOE RSA certificate
    - TOE: Configure "vpn ike-gateway" for test peer with
        1. auth-method :== rsa-signature

2. **remote-id** **:== DN of the IPSec test peer** <u>(excluding NULL ("\0") terminator</u>
(shouldn't match Cert's DN,
can't match SAN)
3. remote-ip :== IPv4 address of IPsec test peer
4. local-id :== FQDN of TOE in the local RSA certificate
5. local-ip :== IPv4 address of TOE
6. key / local-cert : == test_dut _rsa_1.key/.crt(from signed RSA cert/key import)

- TOE: "enable" tunnel to IPsec Test Peer
- Observe: tunnel establishment **FAILS**
- Record: TOE and IPsec Test Peer config settings
- Record: CLI outputs and ACOS Logged Events
- IPsec Test Peer: Restore configuration to saved settings.
- TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that for VPN test peer with FQDN containing a NULL (\0) at the end of the CN certificate that the IPsec tunnel (in particular, the IKE authentication) **FAILS**.
    1. Showed that the value comparison is a bit-wise compare.
    2. DUT will be trying to match the configured remote ID with the subject of the peer certificate.
    3. Whether the peer certificate has or does not have a SAN, does not matter for this test.

- This satisfies the testing requirement.

FCS_NTP_EXT.1 NTP Protocol

<u>**A.2.2.11 FCS_NTP_EXT.1**</u>

FCS_NTP_EXT.1.1 Test #1

FCS_NTP_EXT.1.1

The **version of NTP** selected in element 1.1 and specified in the ST **shall be verified** by <u>observing establishment of a connection to an external NTP server</u> known to be <u>using the specified version(s) of NTP</u>. This may be combined with tests of other aspects of FCS_NTP_EXT.1 as described below.

- Test Steps:
  - NO STEPS: Test uses results from FCS_NTP_EXT.1.4 Test #1

FCS_NTP_EXT.1.2 Test #1a

FCS_NTP_EXT.1.2

The cryptographic algorithms selected in element 1.2 and specified in the ST will have been specified in an FCS_COP SFR and tested in the accompanying Evaluation Activity for that SFR. Likewise, the cryptographic protocol selected in in element 1.2 and specified in the ST will have been specified in an FCS SFR and tested in the accompanying Evaluation Activity for that SFR.

~~[Conditional]~~ ~~**If the message digest algorithm is claimed in element 1.2**, the evaluator will change the message digest algorithm used by the NTP server in such a way that the new value does not match the configuration on the TOE and confirms that the TOE does not synchronize to this time source.~~

- Test Steps:
  - NO STEPS: Message digest algorithm is not claimed in element 1.2 for FCS_NTP_EXT.1.

FCS_NTP_EXT.1.2 Test #1b

The evaluator shall use a packet sniffer to capture the network traffic between the TOE and the NTP server. The evaluator uses the captured network traffic, to

- verify the NTP version,
- to observe time change of the TOE and
- uses the TOE's audit log to determine that the TOE accepted the NTP server's timestamp update.

~~The captured traffic is also used to~~

- ~~verify that the appropriate message digest algorithm was used to authenticate the time source and/or~~
- ~~the appropriate protocol was used to ensure integrity of the timestamp that was transmitted in the NTP packets.~~

- Test Steps:
  - NO STEPS: Test uses results from FCS_NTP_EXT.1.4 Test #1

FCS_NTP_EXT.1.3 Test #1

FCS_NTP_EXT.1.3

The evaluator shall configure NTP server(s) to support periodic time updates to broadcast and multicast addresses. The evaluator shall

- confirm the TOE is configured to not accept broadcast and multicast NTP packets that would result in the timestamp being updated.
- The evaluator shall check that the time stamp is not updated after receipt of the broadcast and multicast packets.

- Test Steps:
  - TOE: "disable" synchronization with all NTP Servers (to be sure)
  - Observe: TOE system time
  - NTP Server #3 Change to enable periodic time update with broadcasts and multicasts. For example:
    1. In 'ntp.conf', add config "broadcastclient" to generate broadcast updates.

        2. In ‘ntp.conf’, add config “broadcastclient 224.0.1.1” to generate multicast updates.
            - Also, enable switch to forward multicasts (224.0.1.1 in particular, if necessary)
    - TOE: “enable” synchronization with NTP Server #1
    - Observe: TOE receiving broadcast and multicast updates through IPsec tunnel.
    - Observe: TOE’s NTP is dropping these broadcast/multicast updates or that TOE’s system time is not changing as a result of these updates.
    - Observe: TOE system time is and not changed to that from NTP Server #3
    - Record: CLI outputs and ACOS Logged Events
    - TOE: “disable” synchronization with NTP Server #1
    - NTP Server #1 Change back to <u>disable</u> periodic time updates with broadcasts and multicasts

- Result:
- **Verdict: Pass**
    - Demonstrated that the TOE does not support NTP broadcast or multicast packet reception/processing.
        1. NOTE: The TOE has does not have any configuration option to enable broadcast or multicast NTP support.

    - Demonstrated that the TOE’s system time is not changed as a result of NTP broadcast or multicast packets directed to the TOE.
- This satisfies the testing requirement.

FCS_NTP_EXT.1.4 Test #1

FCS_NTP_EXT.1.4

Test #1: The evaluator shall confirm the **TOE supports configuration of at least three (3) NTP time sources**. The evaluator shall <u>configure at least three NTP servers</u> to support periodic time updates to the TOE. The evaluator shall confirm the TOE is configured to accept NTP packets that <u>**would result in the timestamp being updated from each of the NTP servers**</u>. The <u>evaluator shall check that the time stamp is updated after receipt of the NTP packets</u>.

The purpose of this test to verify that the TOE can beconfigured to synchronize with multiple NTP servers. It is up to the evaluator to determine that the multi- source update of the time information is appropriate and consistent with the behaviour prescribed by the RFC 1305 for NTPv3 and RFC 5905 for NTPv4.
(TD0528 applied)

- Test Steps:
    - TOE: Save copy OF configuration.
    - TOE: Remove the NTP servers from the TOE configuration.
    - TOE: NTP Status (`show ntp status`) for NTP Servers
    - Observe: No NTP Servers are configured.
    - TOE: Set a new system time (`clock set`)
      (ensure that date/time is different than that used by the NTP Servers)
    - Observe: System time set per manual clock set.

    - TOE: Configure one (1) NTP Server for NTP time synchronization
    - Observe: System time changed due to NTP and the configured NTP server.
    - Observe: ACOS system time is consistent with the with the NTP server.
    - Observe: NTP server becomes synchronized, after a time
    - Record: PCAP of exchanges with the Linux Server as observed on the IPSec GW
    - Record: PCAP of all NTP traffic on the DUT
    - Record: CLI outputs and ACOS Logged Audit/Event records
    - TOE: Remove the NTP server from the TOE configuration.
    - TOE: NTP Status (`show ntp status`) for NTP Servers
    - Observe: No NTP Servers are configured.
    - TOE: Set a new system time (`clock set`)
      (ensure that date/time is different than that used by the NTP Servers)
    - Observe: System time set per manual clock set.
    - TOE: Configure three (3) NTP Servers for NTP time synchronization
    - TOE: Show the status for NTP Servers
    - Observe: All three (3) NTP servers being actively polled (e.g. none indicate ‘unreachable’ status)
    - Observe: System time changed for NTP and the three (3) NTP servers.
    - Observe: ACOS system time is consistent with the with the NTP servers.
    - Observe: One of the three NTP servers (#1) becomes synchronized, after a time.
    - NTP Server (#1): Stop the NTP server service
    - Observe: Another of the two remaining NTP servers (#2) becomes synchronized, after a time.
    - NTP Server (#2): Stop the NTP server service
    - Observe: Last of the three NTP servers (#3) becomes synchronized, after a time.
    - Record: PCAP of exchanges with the Linux Servers as observed on the IPSec GW
    - Record: PCAP of all NTP traffic on the DUT
    - Record: CLI outputs and ACOS Logged Events
    - TOE: Restore configuration to saved settings.

  - This satisfies the testing requirement.

- Result:
- **Verdict: Pass**
    - Demonstrated that the TOE can be configured successfully for time synchronization with all three (3) NTP servers.
    - Demonstrated that the TOE synchronizes its time through all 3 NTP servers, both through logged events and corresponding traffic with the servers (as shown in the PCAP).
    - Demonstrated that TOE’s time stamp is updated after NTP packets are received from the NTP servers configured.
    - In support of FCS_NTP_EXT.1.1 Test 1, this test:
        1. Demonstrated that NTP protocol used is NTP version 4 (NTPv4)
        2. Demonstrated ability to synchronize time using NTPv4 with an external NTP Time server(s) known to be configured for NTP version 4.

  - In support of FCS_NTP_EXT.1.2 Test 1a, this test:
    1. Demonstrated ability to secure communication with NTP servers using IPsec.
  - In support of FCS_NTP_EXT.1.2 Test 1b, this test:
    1. Captured network traffic between the TOE and the NTP server(s).
    2. Verified the NTP version used is NTP Version 4
    3. Demonstrated system time of the TOE is changed from the TOE's internal clock to network time from the NTP server(s)
    4. Demonstrated that the TOE synchronized with the NTP server and that system time of the TOE changed from the internal clock of the TOE to that of the NTP server
    5. Demonstrated that the change to system time is logged to the ACOS event log.
- This satisfies the testing requirement.

FCS_NTP_EXT.1.4 Test #2

FCS_NTP_EXT.1.4 Test #2

~~The evaluator shall also confirm that the **TOE does not synchronize to any other time sources** by configuring another NTP source operating in the server mode and inducing (e.g. send synch request with a forged source IP address) it to send timestamp updates to the TOE. The evaluator shall check that the TSF does not act on the received NTP updates and does not update system time. The evaluator shall confirm that the time stamp is not updated for a valid reason (e.g. failure to recognize NTP authentication key or rejection of unrequested NTP synchronization).~~

Test 2: (The intent of this test is to ensure that the TOE would only accept NTP updates from configured NTP Servers).

The evaluator shall confirm that the **TOE would not synchronize to other, not explicitly configured time sources** by sending an otherwise valid but unsolicited NTP Server responses indicating different time from the TOE's current system time. This rogue time source needs to be configured in a way (e.g. degrade or disable valid and configured NTP servers) that could plausibly result in unsolicited updates becoming a preferred time source if they are not discarded by the TOE. **The TOE is not mandated to respond in a detectable way or audit the occurrence of such unsolicited updates**. The intent of this test is to ensure that the TOE would only accept NTP updates from configured NTP Servers. It is up to the evaluator to craft and transmit unsolicited updates in a way that would be consistent with the behaviour of a correctly-functioning NTP server.

(TD0528 applied)

Note: ~~Double-strikethrough text~~ above indicates content of the former FCS_NTP_EXT.1.5 test that was replaced with FCS_NTP_EXT.1.4 Test #2, per "TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4".

- Test Steps:
  - TOE: "enable" synchronization with NTP Server #1
  - TOE: remove all other NTP server configurations on DUT
  - Observe: TOE system time
  - Observe: TOE synchronized to NTP Server #1 and DUT system time is consistent w/ NTP Server #1.
  - Forging Linux Host : Send synch request with forged source-IP of TOE to NTP Server #2
  - Observe: TOE receiving NTP response from NTP Server #2

- Observe: TOE system time is still consistent w/ NTP Server #1 and not changed to that from NTP Server #2
- Record: PCAP with Forged NTP request packet
- Record: PCAP of NTP traffic to/from NTP Server #1 (should include forged NTP request packet)
- Record: PCAP of NTP traffic to/from NTP Server #2
- Record: CLI outputs and ACOS Logged Events

- Result:
- **Verdict: Pass**
  - Demonstrated that the TOE does not alter system time when it receives an NTP send sync response from an NTP server not otherwise configured on the TOE, such as may be invoked by an NTP send synch request forged from an attacker attempting to disrupt system time on the TOE.

  - Demonstrated that the TOE only reaches out to the configured NTP server with client packets to ask for time update, which does not include any packets set to the NTP server used to forge NTP send sync responses.
- This satisfies the testing requirement.

4.2.3 Identification and Authentication (FIA)

FIA_AFL.1 Authentication Failure Management

FIA_AFL.1

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):

FIA_AFL.1 Test #1

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.

- Test Steps:
  - TOE: Save copy OF configuration.
  - TOE Configure ACOS for maximum # failed login attempts (10) to trigger lockout

```
(admin lockout enable
 admin-lockout threshold 10
 admin lockout duration 15)
```

- FOR EACH of the ACCESS METHODS (SSH CLI, GUI)
    1. Linux Client: login successfully for user "niap-user1" and logout
    2. Linux Client: attempt 10 failed logins for user "niap-user1"

    3. Within subsequent 15 minutes – Perform the following:
        - Observe: 10 failed logins for "niap-user1" events are logged
        - Observe: ACOS lockout for user "niap-user1" is event logged "
        - Linux Client: attempt 11th valid login for user "niap-user1"
        - Linux Client: attempt 12th valid login for user "niap-user1"using other method (e.g. if lockout due to SSH bad logins, try GUI)
        - Observe: attempts rejected @ SSH & @ GUI
        - Observe: failed attempt rejected due to lockout is event logged
        - ACOS Console: attempt 13th valid login for user "niap-user1" and logout
        - Observe: attempt succeeds on ACOS console
        - Linux Client: attempt 14th valid login for user "niap-user1"
        - Observe: failed attempt rejected due to lockout is event logged
        - WAIT 15 MINUTES (for lockout to expire)
        - Linux Client: attempt 15th valid login for user "niap-user1" and logout
        - Observe: attempt succeeds

- Record: CLI outputs and ACOS Logged Events
- TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - Demonstrated ability to detect a configured number of consecutive failed remote logins (e.g. 10) for a given administrator ID (login name)
    - Upon reaching this threshold, demonstrated ability to disable subsequent logins for the given administrator ID (login name) for a configurable, non-zero period of time (e.g. 15 minutes) and confirm that remote logins for both SSH CLI and GUI/HTTPS interfaces are denied, regardless of whether valid credentials are applied.
    - Confirmed that the given administrator ID (login name) can successfully login to the TOE "console" during this lockout period to demonstrate that lockout detection is not applied to the TOE console.
    - After the configured period of time, confirmed that the given administrator ID (login name) can successfully authenticate to the TOE with valid credentials when accessing the TOE with either the SSH CLI or GUI/HTTPS interface.
- This satisfies the testing requirement.

FIA_AFL.1 Test #2

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 Steps:
  - TOE: Save copy OF configuration.
  - TOE: Configure ACOS for maximum # failed login attempts (10) to trigger lockout and change lockout duration to INFINITE

    ```
    (admin lockout enable
     admin-lockout threshold 10
     admin lockout duration 0)
    ```
  - FOR EACH of the ACCESS METHODS (SSH CLI, GUI)
    1. Linux Client: login successfully for user "niap-user1" and logout
    2. Linux Client: attempt 10 failed logins for user "niap-user1"
    3. Observe: 10 failed logins for "niap-user1" events are logged
    4. Observe: ACOS lockout for user "niap-user1" is event logged "
    5. Linux Client: attempt 11th valid login for user "niap-user1"
    6. Linux Client: attempt 12th valid login for user "niap-user1"using other method (e.g. if lockout due to SSH bad logins, try GUI)
    7. Observe: attempts rejected @ SSH & @ GUI
    8. Observe: failed attempt rejected due to lockout is event logged
    9. Linux Client: Login in as "admin"
    10. Linux Client: Attempt unlock user "niap-user1" (`admin niap-user1 unlock`)
    11. Observe: unlock operation FAILS and failure event logged
    12. ACOS Console: Login in as "admin"
    13. ACOS Console: Attempt unlock user "niap-user1" (`admin niap-user1 unlock`)
    14. Observe: unlock operation SUCCEEDS and event logged
    15. Linux Client: attempt 13th valid login for user "niap-user1" and logout
    16. Observe: login attempt succeeds
  - Record: CLI & GUI outputs and ACOS Logged Events
  - TOE: Restore configuration to saved settings.
- Result:
- **Verdict: Pass**
  - Demonstrated ability to detects a configured number of consecutive failed remote logins (e.g. 10) for a given administrator ID (login name)

- Upon reaching this threshold, demonstrated ability to disable subsequent logins for the given administrator ID (login name) until an administrative ‘unlock’ operation is performed and confirm that remote logins for both SSH CLI and GUI/HTTPS interfaces are denied, regardless of whether valid credentials are applied.
- Confirmed that an unlock operation via remote access (e.g. SSH CLI) to the TOE is denied.
- Confirmed that an unlock operation via local console access to the TOE succeeds.
- After the successful unlock operation, confirmed that the given administrator ID (login name) can successfully authenticate to the TOE with valid credentials when accessing the TOE with either the SSH CLI or GUI/HTTPS interface.

- This satisfies the testing requirement.

FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1

The evaluator shall **perform the following tests.**

FIA_PMG_EXT.1 Test #1a

a) Test 1: The evaluator shall *compose passwords that meet the requirements in some way*. For each password, the evaluator shall verify that the TOE supports the password.

- NOTE: tests verify minimum length (8 bytes), maximum length (63 bytes), and all claimed special characters.
- Test Steps:
  - TOE: Save copy OF configuration.
  - FOR EACH of the following passwords for user “niap-user1” CONFIRM they CAN BE configured in ACOS and used to login SUCCESSFULLY

    ```
    Test5678
    TEst !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
    TEst567890123456789012345678901234567890123456789012345678901234567890123
    ```

  - FOR EACH of the following passwords for user “niap-user1” CONFIRM they CANNOT BE configured in ACOS.

    ```
    t
    teST
    teST567
    teST56789012345678901234567890123456789012345678901234567890123456789012345678901234
    ```

  - The following steps are effectively tested in FIA_PMG_EXT.1 Test #1b below and not included in this test case

~~o TOE: Configure ACOS for minimum password length of 15 characters (password-policy min-pswd-len 15)~~

  - ~~FOR EACH of the following passwords for user "niap-user1" CONFIRM they CAN BE configured in ACOS and used to login SUCCESSFULLY~~

    ~~`Test56789012345`~~

  - ~~FOR EACH of the following passwords for user "niap-user1" CONFIRM they CANNOT BE configured in ACOS.~~

    ~~`teST567`~~
    ~~`test5678901234`~~

  - ACOS Record: CLI outputs and ACOS Logged Events
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that in all cases 8-character minimum are supported
  - Demonstrated that 63 characters maximum area supported
  - Demonstrated that all claimed special characters are supported

- This satisfies the testing requirement.

FIA_PMG_EXT.1 Test #1b

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 Steps:
  - TOE: Save copy OF configuration.
  - TOE: Configure ACOS for minimum password length of 15 characters
    (`password-policy min-pswd-len 15`)
  - FOR EACH of the following passwords for user "niap-user1" CONFIRM they CAN BE configured in ACOS and used to login SUCCESSFULLY

    `TEst56789012345`
    `TEst567890123456`
  - FOR EACH of the following passwords for user "niap-user1" CONFIRM they CANNOT BE configured in ACOS.

    `teST5678901234`
  - TOE: Configure ACOS for minimum password length of 63 characters
    (`password-policy min-pswd-len 63`)

  - FOR EACH of the following passwords for user "niap-user1" CONFIRM they CAN BE configured in ACOS and used to login SUCCESSFULLY

TEst567890123456789012345678901234567890123456789012345678901234567890123

- FOR EACH of the following passwords for user “niap-user1” CONFIRM they CANNOT BE configured in ACOS.

  TEst56789012345678901234567890123456789012345678901234567890123456789012

- ACOS Record: CLI outputs and ACOS Logged Events
- TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Test performed here are in addition to FIA_PMG_EXT.1 Test #1a above and further demonstrate range of password lengths supported
  - Demonstrated ability to configure a minimum password length in excess of the default 8 characters and at least 15 characters or greater.
  - Confirmed the passwords at or greater than a configured minimum-password-length are supported.
  - Confirmed the upper bound of 63 bytes as the maximum length that the configured minimum-password-length can be set to
    1. Confired that passwords configured below the upper 63 byte bound are not allowed to be set.

- This satisfies the testing requirement.

FIA_PMG_EXT.1 Test #2

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

- Test Steps:
  - NO STEPS: Test uses results from FIA_PMG_EXT.1 Test #1a (for < 8 char FIPS-mode minimum PW size) and FIA_PMG_EXT.1 Test #1b (for PW minimum sizes set to 15 and 63 char)

FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1

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:

FIA_UIA_EXT.1 Test #1

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.

- Test Steps:
  - TOE: Save copy OF configuration.
  - TOE: Configure for pre-login SSH/console banner (`banner login`) Message saying "SSH – Test Pre-Login Message"
  - TOE: Configure for post-login SSH/console banner (`banner exec`) Message saying "SSH – Test Post-Login Message"
  - TOE: Configure for pre-login GUI banner (GUI @ /gui/#/system/settings/web "Pre GUI Login Message) saying "GUI – Test Pre-Login Message"
  - TOE: Configure for post-login GUI banner (GUI @ /gui/#/system/settings/web) "GUI Login Message saying "GUI – Test Post-Login Message"
  - Observe All banner configuration operations are event logged.

  - FOR EACH of the ACCESS METHODS (SSH CLI and Console)
    1. Linux Client Login to TOE

    2. Observe: "SSH/CONSOLE – Test Pre-Login Message" configured banner is displayed PRIOR TO prompting for Password.
    3. Observe: "SSH/CONSOLE – Test Post-Login Message" configured banner is displayed AFTER successful login.
    4. Observe: In password information is ALWAYS OBFUSCATED (e.g. indicated with characters not echoed)

  - GUI Client: Login to TOE
  - Observe: "GUI – Test Pre-Login Message" configured banner is displayed PRIOR TO prompting for Username/Password.
  - Observe: "GUI – Test Post-Login Message" configured banner is displayed AFTER successful login.
  - Observe: In password information is ALWAYS OBFUSCATED (e.g. indicated with '*' characters or not echoed)
  - FOR EACH of the following method (SSH, Console, GUI)
    1. Login to TOE with BAD USER (e.g. niap-userBAD)
    2. Login to TOE with GOOD USER (e.g. niap-user1) and BAD PASSWORD
    3. Observe: BAD user/password logins FAIL
    4. Observe: Login failures are logged (including the source (ssh, console, GUI))
    5. Observe: In FAILED cases, only the "pre-login banner" and prompts for username/password are the ONLY INFORMATION OF THE DUT DISPLAYED

    6. Login to TOE with GOOD USER (e.g. niap-user1) and GOOD PASSWORD
    7. Observe: GOOD user/password logins SUCCEEDS

8. Observe: Login successes are logged (including the source (ssh, console, GUI))
9. Observe: In BAD/GOOD cases, password information is ALWAYS OBFUSCATED (e.g. indicated with '*' characters in GUI case or not echoed in SSH/Console cases)

  - ACOS Record: CLI outputs and ACOS Logged Events
  - ACOS Record: GUI display of login pages including the pre-login and post-login banner messages.
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated ability to configure banners to be displayed @ CLI (local/remote) and GUI (remote) interfaces BEFORE Identifying and Authentication (I&A) information is input or prompted during administrator login operations to the TOE
  - Demonstrated that incorrect administrator login name and/or password values provided will deny access for all 3 interfaces – SSH-CLI, Console-CLI, GUI-HTTPS.
  - Demonstrated that valid, configured administrator login name and/or password values provided will deny access for all 3 interfaces – SSH-CLI, Console-CLI, GUI-HTTPS.
  - Demonstrated that password information during login operations is obfuscated with '*' characters echoed or no characters echoed during prompted input for all 3 interfaces - SSH-CLI, Console-CLI, GUI-HTTPS.

- This satisfies the testing requirement.

FIA_UIA_EXT.1 Test #2

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.

- Test Steps:
  - TOE: Save copy OF configuration.
  - TOE: Configure for pre-login SSH/console banner (`banner login`) Message saying "SSH – Test Pre-Login Message"
  - TOE: Configure for pre-login GUI banner (GUI @ /gui/#/system/settings/web "Pre GUI Login Message) saying "GUI – Test Pre-Login Message"
  - Linux Client: Connect to TOE via SSH
  - Linux Client: Attempt various escapes (e.g. ^C, ^D, ESC) and various ACOS CLI EXEC commands (e.g. show version), and various Linux commands (e.g. ls -l, cd)
  - Observe: All attempts show nothing more than the pre-login banner and SSH or ACOS login error messages.
  - GUI Client: Connect to TOE via GUI

  - GUI Client: Attempt various escapes (e.g. ^C, ^D, ESC) and various ACOS CLI EXEC commands (e.g. show version), and various Linux commands (e.g. ls -l, cd)
  - Observe: All attempts show nothing more than the pre-login banner and browser or GUI login error messages.
  - ACOS Record: SSH outputs, GUI displays, and ACOS Logged Events
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that no other service or operation is supported by the TOE for REMOTE access (SSH-CLI or GUI-HTTPS) other than the display of a pre-login banner PRIOR to successfully authenticating (logging in) to the TOE.

- This satisfies the testing requirement.

FIA_UIA_EXT.1 Test #3

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.

- Test Steps:
  - TOE: Save copy OF configuration.
  - TOE: Configure for pre-login SSH/console banner (`banner login`) Message saying "SSH – Test Pre-Login Message"
  - ACOS Console: Connect to TOE console
  - ACOS Console: Attempt various escapes (e.g. ^C, ^D, ESC) and various ACOS CLI EXEC commands (e.g. show version), and various Linux commands (e.g. ls -l, cd)
  - Observe: All attempts show nothing more than the pre-login banner and ACOS login error messages.
  - ACOS Record: Console outputs and ACOS Logged Events
  - TOE: Restore configuration to saved settings.

- Expected Result:
- **Verdict: Pass**
  - Demonstrated that no other service or operation is supported by the TOE for LOCAL access (Console-CLI other than the display of a pre-login banner PRIOR to successfully authenticating (logging in) to the TOE.

- This satisfies the testing 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 is not a distributed TOE

FIA_UAU_EXT.2 Password-based Authentication Mechanism

A.2.3.4 FIA_UAU_EXT.2

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.

FCS_ UAU_EXT.2 Test #1

- N/A – Tested in conjunction with FIA_UIA_EXT.1 above.

FIA_UAU.7 Protected Authentication Feedback

A.2.3.5 FIA_UAU.7

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

FIA_UAU.7 Test #1

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 Steps:
  - ○ NO STEPS: Test uses results from FIA_UIA_EXT.1 Test #1 which demonstrated obfuscated passwords during authentication inputs.

FIA_X509_EXT.1 X.509 Certificate Validation

A.2.3.6 FIA_X509_EXT.1/Rev

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/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.:

FIA_X509_EXT.1.1 Test #1a

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 Steps: - Success
  - Create
    1. Self-signed root CA-1 certificate with CA flag TRUE
    2. intermediate CA-2 certificate with CA flag TRUE
    3. Leaf node certificate A issued by CA-2
    4. Leaf node certificate B issued by CA-2
  - Add cert files used here:
  - Configuration:
    1. Configure ipsec gateway A with certificate A and corresponding private key
    2. Install into ipsec gateway A CA-1 and CA-2 certificates
    3. Configure ipsec gateway B with certificate B and corresponding private key
    4. Enable ipsec level 1 debug logs
  - Operation:
    1. Bring up the IPsec tunnel successfully
  - Record the configuration and operational commands and debugs logs showing tunnel is up.
- Result:
- **Verdict: Pass**
  - Demonstrated that when presented with a valid certificate chain (with a peer leaf certification signed by intermediate CA and the intermediate CA signed by the trusted CA) from the VPN peer that the certificate validation function succeeds, and the IPsec tunnel instantiates successfully.
- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #1b

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.

- Test Steps: - – Failure, cert chain does not terminate with trust anchor
  - Set-up

        1. Enable ike-logging-enable configuration option
        2. Remove CA-1 from ipsec gateway A - clear the certificate cache

    - Operation
        1. Enable ike-logging-enable configuration option
        2. Bring up the ipsec tunnel but fail
        3. Enable ipsec debug level 1

    - Record the commands and **failure** reason to bring up the ipsec tunnel

- Result:
- **Verdict: Pass**
    - This test is a continuation of FIA_X509_EXT.1.1 Test #1a above (with a peer leaf certification signed by intermediate CA and the intermediate CA signed by the trusted CA)
    - Demonstrated that removing the trusted (root) CA certificate from the DUT results in validation failure for the certificate presented by the VPN peer and that the IPsec tunnel instantiation **fails**.

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #2

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

- Test Steps:
    - Create cert that is valid until April 1, 2021
    - Verify the current time
    - Set the clock to May 1, 2021
    - Enable ike-logging-enable configuration option
    - Attempt to bring up the tunnel - should fail
    - Record the logs and it should state that cert is expired

- Result:
- **Verdict: Pass**
    - This test is a continuation of FIA_X509_EXT.1.1 Test #2 above
    - Demonstrated that replacing the certificate with one that is expired results in validation failure for the certificate presented by the VPN peer and that the IPsec tunnel instantiation fails.
- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #3a

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.

- Test Steps: - CRL check Leaf Revoked
  - Setup
    1. Create certificate C issued by CA-2 that is revoked
    2. CA-2 issue a CRL with revoked certificate C
    3. CRL distribution point must be in certificate C
  - Configuration
    1. IPsec gateway A with certificate A
    2. IPsec gateway C with certificate C
    3. CA-1 and CA-2 installed in gateway A and C
  - Operation
    1. Enable ike-logging-enable configuration option
    2. Try to bring up tunnel and **fail**
  - Record failure logs and reason. Logs should indicate CRL distribution point is being accessed and certificate C is revoked
- Result:
- **Verdict: Pass**
  a. Demonstrated for a leaf certificate from the VPN peer that CRL indicating revocation of the certificate is detected and the IPsec tunnel instantiation fails
- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #3b

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.

- Test Steps: – OCSP check Leaf Revoked
  - Setup
    1. Create certificate C issued by CA-2 that is revoked
    2. Create OCSP responder certificate issued by CA-2
    3. Configure OCSP responder
    4. OCSP Authority Information Access must be in certificate C
  - Configuration
    1. IPsec gateway A with certificate A
    2. IPsec gateway C with certificate C
    3. CA-1 and CA-2 installed in gateway A and C
  - Operation
    1. Enable ike-logging-enable configuration option
    2. Try to bring up tunnel and **fail**
  - Record failure logs. Logs should indicate OSCP responder and OCSP status is revoked. Capture the packet trace for IKE and OCSP msgs. Show the OSCP status from the CLI
- Result:
- **Verdict: Pass**
  a. Demonstrated for a leaf certificate from the VPN peer that OCSP response indicating revocation of the certificate is detected and the IPsec tunnel instantiation fails
- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #3c

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.

- Test Steps: - OCSP intermediate is revoked
  - Setup
    - Create root CA CA-1
    - Create intermediate CA CA-2 with OCSP Authority Information Access
    - Create intermediate CA CA-3 with OCSP Authority Information Access
    - Create certificate A issued by CA-2
    - Create certificate B issued by CA-3
    - Create certificate C for OCSP service issued by CA-1
    - Revoke CA-2
    - Start OCSP service with cert C
  - Configuration
    - Install CA-1, CA-2 in gateway A
    - IPsec gateway A with certificate A
    - Install CA-1, CA-3 in gateway B
    - IPsec gateway B with certificate B
  - Operation
    - Enable ike-logging-enable configuration option
    - Try to bring up tunnel and **failed**
  - Record OCSP check failure logs of intermediate CA CA-2 on gateway B. Logs should indicate OCSP service is being accessed and validation check failed

- Result:
- **Verdict: Pass**
  a. Demonstrated for an intermediate certificate from the VPN peer that OCSP response indicating revocation of the certificate is detected and the IPsec tunnel instantiation fails

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #3d

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.

- Test Steps: - CRL intermediate is revoked
    - Setup
        - Create root CA CA-1 with CRL Sign key usage
        - Create intermediate CA CA-2 with CRL Distribution Points
        - Create intermediate CA CA-3 with CRL Distribution Points
        - Create certificate A issued by CA-2
        - Create certificate B issued by CA-3
        - Revoke CA-2
        - Generate the CRL by CA-1

    - Configuration
        - Install CA-1, CA-2 in gateway A
        - IPsec gateway A with certificate A
        - Install CA-1, CA-3 in gateway B
        - IPsec gateway B with certificate B

    - Operation
        - Enable ike-logging-enable configuration option
        - Try to bring up tunnel and **failed**

    - Record CRL check failure logs of intermediate CA CA-2 on gateway B. Logs should indicate CRL endpoint is being accessed and validation check failed

- Result:
- **Verdict: Pass**
    a. Demonstrated for an intermediate certificate from the VPN peer that CRL response indicating revocation of the certificate is detected and the IPsec tunnel instantiation fails

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #3-1

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.

- Test Steps:– OCSP check Leaf Not Revoked

- Setup
    1. Start OCSP service with CA-2
    2. Create certificate A and C issued by CA-2
    3. OCSP extension must be in certificate C

- Configuration
    1. IPsec gateway A with certificate A
    2. IPsec gateway C with certificate C
    3. CA-1 and CA-2 installed in gateway A and C

- Operation
    1. Enable ipsec debug level 1 logs
    2. Try to **bring up tunnel, tunnel can up** and we could see the OCSP good log

- Record OCSP good logs. Logs should indicate OCSP service is being accessed and validation check is good

- Result:
- **Verdict: Pass**
    a. Demonstrated for a leaf certificate from the VPN peer that OCSP confirm the certificate as 'not-revoked' and that the IPsec tunnel instantiation **succeeds**

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #3-2

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.

- Test Steps: – CRL check Leaf Not Revoked
    - Setup
        1. Sign the CRL with CA-2
        2. Create certificate A and C issued by CA-2

        3. CRL distribution point must be in certificate C

- ○ Configuration
    1. IPsec gateway A with certificate A
    2. IPsec gateway C with certificate C
    3. CA-1 and CA-2 installed in gateway A and C

- ○ Operation
    1. Enable ipsec debug level 1 logs
    2. Try to **bring up tunnel, tunnel can up** and we could see the CRL is valid log

- ○ Record CRL is valid logs. Logs should indicate CRL service is being accessed and validation check is valid

- Result:
- **Verdict: Pass**

    a. Demonstrated for a leaf certificate from the VPN peer that CRL Distribution Point confirm the certificate as 'not-revoked' and that the IPsec tunnel instantiation **succeeds**

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #3-3

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.

- Test Steps: - OCSP intermediate is not revoked
    - ○ Setup
        - ▪ Create root CA CA-1
        - ▪ Create intermediate CA CA-2 with OCSP Authority Information Access
        - ▪ Create intermediate CA CA-3 with OCSP Authority Information Access
        - ▪ Create certificate A issued by CA-2
        - ▪ Create certificate B issued by CA-3
        - ▪ Create certificate C for OCSP service issued by CA-1

        - Start OCSP service with cert C

    - Configuration
        - Install CA-1, CA-2 in gateway A
        - IPsec gateway A with certificate A
        - Install CA-1, CA-3 in gateway B
        - IPsec gateway B with certificate B

    - Operation
        - Enable ipsec debug level 1 logs
        - Try to **bring up tunnel and succeed**

    - Record OCSP check logs of intermediate CA. Logs should indicate OCSP service is being accessed and validation check succeed

- Result:
- **Verdict: Pass**
    a. Demonstrated for an intermediate certificate from the VPN peer that OCSP confirm the certificate as 'not-revoked' and that the IPsec tunnel instantiation **succeeds**

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #3-4

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.

- Test Steps: - CRL intermediate is not revoked
    - Setup
        - Create root CA CA-1 with CRL Sign key usage
        - Create intermediate CA CA-2 with CRL Distribution Points
        - Create intermediate CA CA-3 with CRL Distribution Points
        - Create certificate A issued by CA-2
        - Create certificate B issued by CA-3
        - Generate the CRL by CA-1

- Configuration
    - Install CA-1, CA-2 in gateway A
    - IPsec gateway A with certificate A
    - Install CA-1, CA-3 in gateway B
    - IPsec gateway B with certificate B

- Operation
    - Enable ipsec debug level 1 logs
    - Try to **bring up tunnel and succeed**

- Record CRL check logs of intermediate CA. Logs should indicate CRL endpoint is being accessed and validation check succeed

- Result:
- **Verdict: Pass**
    a. Demonstrated for a intermediate certificate from the VPN peer that CRL Distribution Point confirm the certificate as 'not-revoked' and that the IPsec tunnel instantiation **succeeds**

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #4-1

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

If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

- Test Steps: – OCSP check
    - Setup
        1. Create certificate D issued by CA-2 that does not have the OCSP Signing key usage.
        2. Start OCSP service with cert D
        3. Create certificate C issued by CA-2
        4. OCSP extension must be in certificate C

    - Configuration
        1. IPsec gateway A with certificate A
        2. IPsec gateway C with certificate C
        3. CA-1 and CA-2 installed in gateway A and C

    - Operation
        1. Enable ike-logging-enable configuration option
        2. Try to bring up tunnel and **tunnel can up but** we **see the failure log**

    - Record OCSP failure logs, and reason. Logs should indicate OCSP service is being accessed and validation check failed
    -
- Result:
- **Verdict: Pass**
    - Demonstrated that for an OCSP certificate response that does not have an OCSP signing purpose results in validation failure for an OCSP response and is treated by the TOE as if the OCSP server couldn't be contacted or otherwise indicated a failure in the OCSP response.

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #4-2

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

If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

- Test Steps: – CRL check
    - Setup
        1. Create certificate E issued by CA-2 that does not have the cRLsign key usage.
        2. Sign the CRL with cert E
        3. Create certificate C issued by CA-2
        4. CRL distribution point must be in certificate C

    - Configuration
        1. IPsec gateway A with certificate A
        2. IPsec gateway C with certificate C
        3. CA-1 and CA-2 installed in gateway A and C

    - Operation
        1. Enable ike-logging-enable configuration option
        2. Try to bring up tunnel and **tunnel can up but** we **see the failure log**

    - Record CRL failure logs, and reason. Logs should indicate CRL service is being accessed and validation check failed

- Result:
- **Verdict: Pass**

    - Demonstrated that a CRL without the “cRLsign key usage” bit set is considered invalid and is treated by the TOE as if the CRL distribution point couldn’t be contacted or did not have a corresponding revocation entry.

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #5

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.)

- Test Steps:
    - Setup
        1. Use certificates A and B
    - Configuration
        1. Modify a byte in the first 8 bytes of the certificate A
    - Operation
        1. Enable ike-logging-enable configuration option
        2. Connection should **fail**
    - Show no SAs established and record ACOS event logs and error

- Result:
- **Verdict: Pass**
    - Demonstrated modification of the ASN.1 SEQUENCE encodings in the first 8-bytes provided by the VPN peer certificate causes the certificate validation to fail. Fundamental/basic ASN.1 parsing of the certificate will fail and the IPsec tunnel instantiation fails.

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #6

f) Test 6: The evaluator shall modify any byte in the certificate signatureValue field (see RFC5280 Sec. 4.1.1.3), which is normally the last field in the certificate, and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

- Test Steps:
    - Setup
        1. Use certificates A and B
    - Configuration
        1. Modify the last byte in the certificate A
    - Operation
        1. Enable ike-logging-enable configuration option
        2. Connection should **fail**

    - Show no SAs established and record the ACOS event logs and errors

- Result:
- **Verdict: Pass**
    - Demonstrated modification of the certificate's 'signatureValue' field corrupts the VPN peer certificate's signature and causes the certificate validation to fail, resulting in failure of the IPsec tunnel to instantiate.

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #7

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.)

- Test Steps:
    - Setup
        1. Enable ike-logging-enable configuration option
        2. Use certificates A and B
    - Configuration
        1. Modify the public key in Certificate A
    - Operation
        1. Enable ike-logging-enable configuration option
        2. Connection should **fail**
    - Show no SAs established and record the IKE logs and errors
- Result:
- **Verdict: Pass**
    - Demonstrated modification of the certificate's 'public key' field corrupts the VPN peer certificate's integrity and causes the certificate validation to fail. Fundamental hashing integrity of the certificate will fail and the IPsec tunnel instantiation fails.

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #8a

From: TD0527 - Updates to Certificate Revocation Testing (FIA_X509_EXT.1)

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 Steps:
  - Create
    1. Self-signed root ECDSA CA-1 certificate with CA flag TRUE
    2. intermediate ECDSA CA-2 certificate with CA flag TRUE
    3. Leaf node ECDSA certificate A issued by CA-2, add the CA cert chain (include CA-1 and CA-2) to A
    4. Leaf node ECDSA certificate B issued by CA-2, add the CA cert chain (include CA-1 and CA-2) to B
  - Add cert files used here:
  - Configuration:
    1. Configure ipsec gateway A with certificate A and corresponding private key
    2. Install into ipsec gateway A CA-1 certificate as CA cert
    3. Configure ipsec gateway B with certificate B and corresponding private key
    4. Install into ipsec gateway B CA-1 certificate as CA cert
    5. Enable ipsec level 1 debug logs
  - Operation:
    1. Bring up the IPsec tunnel **successfully**
  - Record the configuration and operational commands and debugs logs showing tunnel is up.
- Result:
- **Verdict: Pass**
  - Demonstrated that the TOE supports an Elliptical Curve (EC) certificate chain from the VPN peer, including leaf and intermediate certificates.
  - Demonstrated that the TOE supports named curves in valid EC certificates from the VPN Peer and that the IPsec tunnel instantiation successfully.

- This satisfies the testing requirement.

FIA_X509_EXT.1.1 Test #8b

From: TD0527 - Updates to Certificate Revocation Testing (FIA_X509_EXT.1)

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 Steps:
  - Create
    - Create ECDSA root CA CA-1
    - Create ECDSA intermediate CA CA-2 with named curve
    - Create ECDSA intermediate CA CA-3 with explicit EC parameter
      1. Use OpenSSL's command line tool to convert the EC key generated from a named curve to an explicit curve:

```
openssl ec -in ca_2.key -param_enc explicit -out ca_3_explicit.key
```

    - Create certificate A issued by CA-2
    - Create certificate B issued by CA-3
  - Configuration:
    - Install CA-1, CA-2 in gateway A (DUT)
    - IPsec gateway A with certificate A
    - Install CA-1, CA-3 in gateway B
    - IPsec gateway B with certificate B
  - Operation:
    - Enable ike-logging-enable configuration option
    - Bring up the IPsec tunnel **failed**
  - Record the configuration and operational commands and ACOS event logs showing tunnel is not up.
- Result:
- **Verdict: Pass**
  - Demonstrated that the TOE detects the use of "explicit" elliptical curves in an intermediate certificate as part of an EC certificate chain from the VPN Peer, rejects the certificate as invalid, and that the IPsec tunnel instantiation fails.

  - This satisfies the testing requirement

FIA_X509_EXT.1.1 Test #8c-1

From: TD0527 - Updates to Certificate Revocation Testing (FIA_X509_EXT.1)

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.

- Test Steps:
  - Create
    - Self-signed root ECDSA CA-1 certificate with CA flag TRUE
    - intermediate ECDSA CA-2 certificate with CA flag TRUE
    - Leaf node ECDSA certificate A issued by CA-2
    - Leaf node ECDSA certificate B issued by CA-2
  - Add cert files used here:
  - Configuration:
    - Enable ike-logging-enable configuration option
    - Configure ipsec gateway A with certificate A and corresponding private key
    - Install into ipsec gateway A CA-1 certificate
    - Configure ipsec gateway B with certificate B and corresponding private key
    - Install into ipsec gateway B CA-1 certificate
    - Enable ipsec level 1 debug logs
  - Operation:
    - Bring up the IPsec tunnel **successfully**
  - Record the configuration and operational commands and debugs logs showing tunnel is up.
- Result:
- **Verdict: Pass**
  - Demonstrated that the TOE supports an Elliptical Curve (EC) certificate chain loaded onto the TOE, including leaf and intermediate certificates.
  - Demonstrated that the TOE supports named curves in valid EC certificates from these certificates and that the IPsec tunnel instantiation successfully
- This satisfies the testing requirement

FIA_X509_EXT.1.1 Test #8c-2

From: TD0527 - Updates to Certificate Revocation Testing (FIA_X509_EXT.1)

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.

- Test Steps:
  - Create
    - Self-signed root ECDSA CA-1 certificate with CA flag TRUE
    - intermediate ECDSA CA-2 certificate with CA flag TRUE

      1. Use OpenSSL's command line tool to convert the EC key generated from a named curve to an explicit curve:

         openssl ec -in ca_2.key -param_enc **explicit** -out ca_2_explicit.key

    - Then generate CA-2 cert with the explicit curve key

    - Leaf node ECDSA certificate A issued by CA-2
    - Leaf node ECDSA certificate B issued by CA-2

  - Add cert files used here:

  - Configuration:
    - Configure ipsec gateway A with certificate A and corresponding private key
    - Install into ipsec gateway A CA-1 and CA-2 certificate
    - Configure ipsec gateway B with certificate B and corresponding private key
    - Install into ipsec gateway B CA-1 and CA-2 certificate

  - Operation:
    - Enable ike-logging-enable configuration option
    - Bring up the IPsec tunnel **failed**

  - Record the configuration and operational commands and ACOS event logs showing tunnel is up.

- Result:
- **Verdict: Pass**

  - Demonstrated that the TOE detects the use of "explicit" elliptical curves in an intermediate certificate as part of an EC certificate chain configured on the TOE and makes the file unavailable for loading into the TOE's trust store on its initial use in establishing an IPsec tunnel.

  - This satisfies the testing requirement

FIA_X509_EXT.1.2 Test #1

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).

- Test Steps:
    - Create
        1. intermediate CA-3 certificate without basicConstraints CA flag
        2. Leaf node certificate A issued by CA-1
        3. Leaf node certificate D issued by CA-3
    - Add Cert files used here:
    - List ACOS commands used to set and configure the certs
    - Enable ike-logging-enable configuration option
    - Record the **failure** (either import CA cert, fail to configure it in ipsec gateway, or fail to bring ipsec tunnel)

- Result:
- **Verdict: Pass**
    - Demonstrated that the TOE detects the "basicConstraints extension" missing in an intermediate certificate uploaded on the TOE and makes the file unavailable for loading into the TOE's trust store on its initial use in establishing an IPsec tunnel.

- This satisfies the testing requirement

FIA_X509_EXT.1.2 Test #2

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 Steps:
  - Create
    1. intermediate CA-4 certificate with basicConstraints CA flag set to FALSE
    2. Leaf node certificate A issued by CA-1
    3. Leaf node certificate D issued by CA-4

  - Add Cert files used here:
  - List ACOS commands used to set and configure the certs
  - Enable ike-logging-enable configuration option
  - Record the **failure** (either import CA cert, fail to configure it in ipsec gateway, or fail to bring ipsec tunnel)

- Result:
- **Verdict: Pass**
  - Demonstrated that the TOE detects the "basicConstraints extension" set to FALSE in an intermediate certificate uploaded on the TOE and makes the file unavailable for loading into the TOE's trust store on its initial use in establishing an IPsec tunnel.

- This satisfies the testing requirement.

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2

The evaluator **shall perform the following test for each trusted channel**:

FIA_X509_EXT.2.1 Test #1

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.~~

- Test Steps: - Success
  - Set-up
    1. Shutdown the OCSP responder
  - Configuration:
    1. Use certs A and C
  - Operation:
    1. Bring tunnel up
  - Logs and packet capture should show OCSP responder not reachable
  - **Tunnel should come up**

- Result:
- **Verdict: Pass**
  - Demonstrated that when the TOE cannot access OCSP and CRL Distribution Points referenced by a certificate presented to the TOE during IPsec tunnel establishment that the TOE considers the certificate as not revoked and will accept the certificate, allowing the IPsec tunnel to establish successfully.

- This satisfies the testing requirement.

FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3

The evaluator shall **perform the following tests:**

FIA_X509_EXT.3.1 Test #1

a) **Test 1:** The evaluator shall use the guidance documentation to cause the TOE to generate a Certification Request. The evaluator shall capture the generated message and ensure that it conforms to the format specified.

The evaluator shall confirm that the Certification Request provides the public key and other required information, including any necessary user-input information.

- Test Steps:
  - Create
    1. RSA CSR with all fields CN, C, O, OU above
       - CSR should also include Locality, State/Province, and email fields
       - CSR must indicate 2048 bit key size
    2. ECDSA CSR with all fields CN, C, O, OU above
       - CSR should also include Locality, State/Province, and email fields

        - CSR must indicate 256 or 384 bit key size

    - Show the encoded and decoded CSRs

- Result:
- **Verdict: Pass**
    - Demonstrated that TOE generation of a Certification Request (aka Certificate Signing Request (CSR)) supports the requisite Common Name (CN), Organization (O), Organizational Unit (OU), and Country (C) values/parameters.
    - Demonstrated that TOE generation of a Certification Request (aka Certificate Signing Request (CSR)) supports the following device-specific information values/parameter indicated in the Guidance document
        1. Locality, State/Province, email.

    - Demonstrated CSRs generated with
        1. RSA: 2048 bit key
        2. ECDSA: 256 or 384 bit key
- This satisfies the testing requirement.

FIA_X509_EXT.3.2 Test #1

b) **Test 2**: The evaluator shall demonstrate that validating a response message to a Certification Request without a valid certification path results in the function failing.

The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the certificate response message and demonstrate that the function succeeds

- Test Steps:
    - Create
        - Self-signed root ECDSA CA-1 certificate with CA flag TRUE
        - intermediate ECDSA CA-2 certificate with CA flag TRUE
        - Leaf node ECDSA certificate A issued by CA-2
        - Leaf node ECDSA certificate B issued by CA-2 (from ECDSA CSR generated by the DUT)

    - Add cert files used here:
        - CA-1 named ca.crt
        - CA2 named interca3.crt
        - Certificate A and private key A named ec-ax2.crt/ec-ax2.key
        - Certificate B and private key B named ec-ax.crt/ec-ax.key

    - Configuration:
        - Enable ike-logging-enable configuration option
        - Configure ipsec gateway A with certificate A and corresponding private key
        - Install into ipsec gateway A CA-1 certificate

    - Configure ipsec gateway B with certificate B and corresponding private key
    - Install into ipsec gateway A intermediate ECDSA CA-2 certificate with CA flag

  - Operation:
    - Bring up the IPsec tunnel **fails** (Root CA not found on DUT for intermediate cert provided by peer)

  - Configuration:
    - Remove into ipsec gateway A intermediate ECDSA CA-2 certificate with CA flag TRUE
    - Install into ipsec gateway B intermediate ECDSA CA-2 certificate with CA flag TRUE

  - Operation:
    - Bring up the IPsec tunnel **fails** (since root-CA cert not found on DUT & not provided by peer)

  - Configuration:
    - Install into ipsec gateway B CA-1 certificate

  - Operation:
    - Bring up the IPsec tunnel s**ucceeds**

- Result:
- **Verdict: Pass**
  - Demonstrated that when the TOE is presented with IPsec Certificate response information by the IPsec peer without a valid (or complete) certification path of certificates that the certificate validation fails and the IPsec tunnel is not instantiated.
  - Demonstrated further that when missing elements (e.g. intermediate certificates) are configured on the TOE that they complete the certification path and that certificate validation succeeds and the IPsec tunnel is instantiated.

- This satisfies the testing requirement.

4.2.4 Security Management (FMT)

FMT_MOF.1/ManualUpdate Management of security functions behaviour

FMT_MOF.1(1)/ManualUpdate

FMT_MOF.1/ManualUpdate Test #1

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

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.

- Test Steps:
  - TOE: Save copy OF configuration.
  - Linux Client: Logon as “niap-user1” on TOE “
  - Linux Client: Attempt ACOS trusted update (using digital signature) - primary
  - Linux Client: Attempt ACOS trusted update (using digital signature) - secondary
  - Observe: Updates SUCCEEDS and are event logged
  - Linux Client: Logon as “niap-ro+sys-user2” on TOE “
  - Linux Client: Attempt ACOS trusted update (using digital signature) - primary
  - Observe: Update FAILS and are log event logged (including reason for failure as insuff privilege or something similar)
  - Linux Client: Logon as “niap-rw+part-user3” on TOE “
  - Linux Client: Attempt ACOS trusted update (using digital signature) - primary
  - Observe: Update FAILS and are log event logged (including reason for failure as insuff privilege or something similar)
  - ACOS Record: CLI outputs and ACOS Logged Events
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that TOE administrators without sufficient privilege fail in attempts to update the TOE . This includes TOE administrators configure as either:
    1. Read-only privileged
    2. Partition privileged.

  - Demonstrated that TOE security administrators are allowed to successfully update the TOE . This includes TOE administrators configure with “system (e.g. non-partition specific)” and “read+write” privilege.
    1. Both primary and secondary images updates to the TOE are demonstrated.

- This satisfies the testing requirement.

FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1/CoreData

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.

FMT_MTD.1/CoreData Test #1a

- Test Steps:
    - TOE: Save copy OF configuration.
    - Linux Client: Logon as “niap-ro+sys-user2” on TOE “
    - TOE: Confirm that the following ACOS configuration TSF-data related CLI commands are not recognized for such administrator **without** “write privilege"
        - access-list
        - admin
        - admin-lockout
        - banner
        - bootimage
        - hostname
        - interface
        - ip
        - no
        - ntp
        - pki
        - system
        - terminal idle-timeout 10
        - timezone
        - upgrade
        - vpn
        - web-service
    - ACOS Record: CLI outputs and ACOS Logged Audits/Events
    - TOE: Confirm that the following ACOS configuration TSF-data related CLI commands are rejected/failed due to access denial (or similar error) for such administrator **without** “write privilege"
        - clear audit
        - clear logging
        - clock set
        - configure
        - export ca_cert
        - export cert
        - export cert-key
        - export crl
        - export csr
        - export key
        - import aflex
        - import auth-portal
        - import auth-portal-image
        - import auth-saml-idp
        - import bw-list

        - import ca_cert
        - import cert
        - import cert-key
        - import class-list
        - import class-list-convert
        - import crl
        - import dnssec-dnskey
        - import dnssec-ds
        - import geo-location
        - import glm-cert
        - import glm-license
        - import health-external
        - import health-postfile
        - import ip-map-list
        - import key
        - import local-uri-file
        - import lw-4o6
        - import policy
        - import rpz
        - import store
        - import thales-secworld
        - import usb-license
        - import web-categorylicense
        - import wsdl
        - import xml-schema
        - reboot
        - write memory

    - ACOS Record: CLI outputs and ACOS Logged Audits/Events

    - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - Demonstrated that non-Security Administrators (e.g. admins without write privilege cannot perform management operations that affect (e.g. change/modify) TSF data.

- This satisfies the testing requirement.

FMT_MTD.1/CoreData Test #1b

- Test Steps:
  - TOE: Save copy OF configuration.

  - Linux Client: Logon as “niap-user1” on TOE “
  - TOE: Confirm that the following ACOS configuration TSF-data related CLI commands are supported for such a “write privilege" administrator (e.g a Security Admin)
    - access-list
    - configure
    - hostname
    - interface
    - ip address
    - ip control-apps-use-mgmt-port
    - write memory
    - reboot

  - ACOS Record: CLI outputs and ACOS Logged Audits/Events
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that Security Administrators (e.g. admins WITH write privilege perform management operations that affect (e.g. change/modify) TSF data not otherwise demonstrated in other testing activities.

- This satisfies the testing requirement.

FMT_MTD.1/CoreData Test #1c

- Test Steps:
  - TOE: Save copy OF configuration.
  - ACOS CLI Client: Show VPN IKE GW & IPSec GW config.
  - ACOS CLI Client: Disable VPN Event logging (to minimize clutter/volume in Event Log)
  - **ACOS CLI Client: Add a new ACOS read-write admin account**
  - ACOS CLI Client: Change the password for the read-write admin account
  - **ACOS CLI Client: Delete the added read-write admin accounts**
  - ACOS GUI Client: Add a new ACOS read-write admin account
  - **ACOS GUI Client: Add a new ACOS read-only admin account**
  - **ACOS GUI Client: Change the password for the read-only admin account**
  - **ACOS GUI Client: Delete the added read-only admin accounts**
  - **ACOS GUI Client: Set the ACOS CLI and GUI idle timeouts (… may be two different operations)**

- **ACOS GUI Client: Set the Admin Lockout parameters (duration, threshold, & enable … *if available*)**
- ACOS GUI/CLI: Trigger admin lockout for added read-write admin account added above
- ACOS GU Client I: Unlock the locked out admin account
- **ACOS GUI Client: Remove 3 configured NTP servers**
- **ACOS GUI Client: Added the 3 NTP servers back to the DUT config**
- ACOS GUI Client: Add BYPASS ACL to allow GUI access to DUT's local port 443
- ACOS GUI Client: Login via direct connection to the DUT (using this BYPASS ACL)
- ACOS GUI Client: Remove IKE GW and IPSec configuration for IPSec tunnel
- Observe: Can no longer ping NTP servers
- **ACOS GUI Client: Add IKE GW and IPSec configuration for IPSec tunnel**
- Observe: Can now ping NTP servers
- ACOS GUI Client: Logout from direct connection to the DUT
- ACOS GUI: Remove BYPASS ACL to allow GUI access to DUT's local port 443
- **ACOS GUI Client: Perform a file import operation (e.g. AFLEX file)) using SCP or SFTP**
- ACOS GUI Client: Perform a file export operation (e.g. AFLEX file)) using SCP or SFTP
- **ACOS GUI Client: Perform an ACOS upgrade operation – primary image**
- **ACOS GUI Client: Perform an ACOS upgrade operation – secondary image**
- ACOS CLI Client: Re-enable VPN Event logging.
- Record: GUI snapshots and ACOS audit and logged Events
- TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated summary ability to configure the TOE using ACOS GUI management interface operations (via IPsec) comparable to ACOS CLI operations demonstrated elsewhere in this test plan and where ACOS GUI management steps were not included in the procedures.
  - Also demonstrated some miscellaneous other ACOS CLI configuration operations for convenience of sample log-entry collection.
    1. Add/delete/change-password for administrator user account

- This satisfies the testing requirement.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1

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.

FMT_SMF.1 Test #1

- Test Steps:
  - NO STEPS: No separate testing needed as all necessary management functions in in FMT_SMF.1.1 are exercised in other test activities.

FMT_SMR.2 Restrictions on security roles

FMT_SMR.2

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.

FMT_SMR.2 Test #1

- Test Steps:
  - NO STEPS: No separate testing needed as all supported interfaces are exercised in other test activities.

FMT_MOF.1/Functions Management of security functions behavior

FMT_MOF.1/Functions

FMT_MOF.1 Test #1.1

~~**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.~~

- N/A - 'transmission of audit data to external IT entity' not claimed for FAU_MOF.1/Functions

FMT_MOF.1 Test #1.2

~~**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.~~

- ○ N/A - 'transmission of audit data to external IT entity' not claimed for FAU_MOF.1/Functions

FMT_MOF.1 Test #1.3

~~**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.~~

- ○ N/A - 'handling of audit data' not claimed for FAU_MOF.1/Functions

FMT_MOF.1 Test #1.4

~~**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.~~

- ○ N/A - 'handling of audit data' not claimed for FAU_MOF.1/Functions

FMT_MOF.1 Test #1.5a

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 Steps:
  - ○ TOE: Save copy OF configuration.
  - ○ *TOE: Generate enough events to completely fill the ACOS Local Log*
  - ○ *ACOS Record: Show that oldest log entries are removed from the log as new entries are being added.*
  - ○ TOE Try to modify config without authentication (take screenshot of login page to show cannot access config without authentication)

- TOE: login with read-only user niap-user1.
- TOE: try to modify *audit/logging configuration settings* (user will not be allowed in config mode)
- ACOS Record: CLI outputs and ACOS Logged Events

- Result:
- **Verdict: Pass**

  - *Demonstrated that the ACOS local log is full and is overwriting oldest log entries with new ones.*

  - Demonstrated that prior to authentication no commands or operations can be performed on the TOE.
    1. @ CLI only the authentication can be entered
       - Attempting to enter CLI commands is treated as simply bad passwords and eventually fails the authentication due to 3 passwords allowed in an attempt.
    2. @ GUI only authentication fields are available
       - Attempting to enter CLI commands is treated as simply bad usernames or passwords and eventually fails the redraws the login page and fails the authentication after 3 bad attempts.

  - Demonstrated that an unprivileged (e.g. read-only) administrator cannot perform commands that affect ACOS event logging or audit logging.

- This satisfies the testing requirement.

FMT_MOF.1 Test #1.5b

- Test Steps:

  - TOE: Create user niap-user1 with read only access
  - TOE: login to ACOS with niap-user1

  - TOE: Try to modify audit/event log config.
  - Observe: DUT should not allow user to get into config mode
  - Observe: user should only be able to view local syslog/audit log

  - ACOS Record: CLI outputs and ACOS Logged Events
  - TOE: Try to access the box without user login (screenshot of login page to show we cannot get to the config page)

- Result:
- **Verdict: Pass**

    - Demonstrated that an unprivileged (e.g. read-only) administrator cannot perform commands that affect ACOS event logging or audit logging used to configure handling of ACOS audit/event logs (e.g. audit-data).
        - See FAU_MOF.1 Test #1.5a for this restriction, not repeated in this test case.

- This satisfies the testing requirement.

FMT_MOF.1 Test #1.6

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 Steps:

```
NIAP-NDv22e-DUT1(config)#audit ?
  enable   Enable audit service
  size     Config audit buffer size, default is 20,000  size    Config
audit buffer size, default is 20,000
NIAP-NDv22e-DUT1(config )#logging buffered ?
  <10000-50000>   Logging buffer size (in items), default 30000
  disable         Do not send log to buffer
  emergency       System unusable log messages      (severity=0)
```

    - TOE: Create user admin with read/write access
    - TOE: login to ACOS with admin.
    - ACOS DUT Show no audit log server is configured
    - ACOS DUT Try to configure (add) an audit log server to the ACOS configuration.
    - Observe: DUT should allow user to configure audit server
    - *Observe: Modified settings (with CLI 'show' commands)*
    - Observe: Logs are sent to the added audit log server.
    - ACOS Record: CLI outputs and ACOS audit and logged Events
    - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - Demonstrated that an authenticated security administrator (read+write system admin)
        1. Can successfully perform CLI commands affecting audit server related configuration.
        2. @ GUI show that these same configuration settings can be set.

    - *Demonstrated that modified logging configuration parameters are in effect.*

- This satisfies the testing requirement.

FMT_MOF.1 Test #1.7

~~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.~~

  - N/A - 'determine the behaviour of' not claimed for FAU_MOF.1/Functions

FMT_MOF.1 Test #1.8

~~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.~~

  - N/A - 'determine the behaviour of' not claimed for FAU_MOF.1/Functions

FMT_MTD.1/ CryptoKeys Management of TSF data

FMT_MTD.1/CryptoKeys

FMT_MTD.1. Test #1

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.

- Test Steps:
  - TOE: Save copy OF configuration.

- TOE: Try to modify public and private keys config without authentication (take screenshot of login page to show cannot access config without authentication)
- TOE: login with read-only user niap-user1.
- TOE: try to modify public and private key configuration (user will not be allowed in config mode)
- ACOS Record: CLI outputs and ACOS Logged Events
- TOE: Restore configuration to saved settings.
- TOE: Create user admin with read/write access
- TOE login to ACOS with admin.
- TOE Try to modify public and private key config.
- Observe: DUT should allow user to modify public and private key   configuration as well as delete ,import or export them
- ACOS Record: CLI outputs and ACOS Logged Events
- TOE: Restore configuration to saved settings

- Result:
- **Verdict: Pass**
  - Demonstrated that prior to authentication no commands or operations (including those that manage crypto keys) can be performed on the TOE.
    1. @ CLI only the authentication can be entered
       - Attempting to enter CLI commands is treated as simply bad passwords and eventually fails the authentication due to 3 passwords allowed in an attempt.
    2. @ GUI only authentication fields are available
       - Attempting to enter CLI commands is treated as simply bad usernames or passwords and eventually fails the redraws the login page and fails the authentication after 3 bad attempts.
  - Demonstrated that an unprivileged (e.g. read-only) administrator cannot perform commands that manage crypto keys (including creation, deletion, import, and export).
  - Demonstrated that an authenticated security administrator (read+write system admin) can indeed perform commands that manage crypto keys (including CSR generation, CSR export, Certificate imports (signed TOE cert from CSR, intermediate cert, root-CA cert)
    1. @ CLI and @ GUI CAN create a certificate
    2. @ CLI can
       - generate CSR, export CSR,
       - import certificates (signed from CSR, intermediate certs, root-CA)
       - delete CSR, certs (one or more)
    3. @ GUI can
       - generate CSR, export CSR,
       - import certificates (signed from CSR, intermediate certs, root-CA)
       - delete CSR, certs (one or more)

- This satisfies the testing requirement.

4.2.5 Protection of the TSF (FPT)

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

FPT_SKP_EXT.1

Tests N/A

FPT_SKP_EXT.1 Test #1

- N/A – No testing activities are defined

FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1

Tests N/A

FPT_APW_EXT.1 Test #1

- N/A – No testing activities are defined

FPT_TST_EXT.1 TSF Testing (Extended)

FPT_TST_EXT.1

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

FPT_TST_EXT.1 Test #1

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

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

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.~~

- Test Steps:
  - TOE: Save copy OF configuration.
  - TOE: Reboot the DUT in fips mode.
  - ACOS Record: From shell get kernel crypto engine logs.
  - ACOS Record: From var log verify openssl library passed POST.

    ```
    show varlog | inc FIPS library power on self-test passed.
    ```
  - ACOS Record: From varlog verify bootup integrity hash check.

    ```
    show varlog | inc Image verification check passed.
    ```
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - demonstrated self test at system start-up (boot) of the Linux kernel crypto engine (provider) succeeds (even for algorithms not in the scope of the TOE).

- This satisfies the testing requirement.

o The TOE is not a distributed TOE

FPT_TST_EXT.1 Test #2

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

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

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.~~

- Test Steps:
  - TOE: modify /boot/signature.pri file (to simulate image integrity failure) from development shell
  - TOE: reboot to the primary image
  - Observe: Observe that the DUT boots up and indicates FIPS FAILURE MODE when logged logged into the DUT's console.
  - Observe: Observe that the varlog indicates failure due to "Image verification check failed".
  - TOE: 'show time' (to show system time is close to the varlog entry)
  - Observe: Remote management services (SSH, HTTP, HTTPS) are not available on the DUT's management port or that the DUT's management port is off-line.
  - Observe: All DUT data ports are off-line (`show interfaces`).
  - ACOS Record: Observations above from the DUT console and remote SSH/GUI applications.

  - TOE: In FIPS FAILURE MODE, reboot to the secondary (good) image
  - TOE: Upgrade the DUT primary image in fips mode with the FAILED BUILD IMAGE for failed algorithm tests and reboot to DUT primary.
  - Observe: Observe that the DUT boots up and indicates FIPS FAILURE MODE when logged logged into the DUT's console.
  - Observe: Observe that the varlog indicates failure due to "FIPS library power on self test failed".
  - TOE: 'show time' (to show system time is close to the varlog entry)
  - Observe: Remote management services (SSH, HTTP, HTTPS) are not available on the DUT's management port or that the DUT's management port is off-line.
  - Observe: All DUT data ports are off-line.
  - ACOS Record: Observations above from the DUT console and remote SSH/GUI applications.
  - TOE: Reboot the DUT secondary image.

- Result:
- **Verdict: Pass**
  - Demonstrated self test failure at system start-up (boot) of the OpenSSL crypto provider is detected.
  - Demonstrated integrity tests failure at system start-up (boot) of the device is detected

- This satisfies the testing requirement.
- The TOE is not a distributed TOE

FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1

The evaluator shall **perform the following tests**:

FPT_TUD_EXT.1 Test #1

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.

- Test Steps:

TOE:	Save copy OF configuration.
TOE:	show version of ACOS & confirm older version
HTTPS/SFTP/SCP Server	download valid update (test revision) for update by ACOS
TOE:	update ACOS from the valid update – primary image
TOE:	update ACOS from the valid update – secondary image
TOE:	reboot – to primary image
TOE:	reboot – to secondary image
Observe:	all update operations SUCCEDED and logged events of the upgrade
Observe:	all reboots SUCCEEDED
Observe:	Updated ACOS version after all reboots
ACOS Record:	CLI outputs and ACOS Logged Events
TOE:	Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - Demonstrated TOE's ability to show the current installed versions of ACOS, both for primary and secondary installed images.
    - Demonstrated TOE ability to update the primary image with immediate reboot and show that the image is updated

- Demonstrated TOE ability to update the secondary image with immediate reboot and show that the image is updated
- Demonstrated TOE ability to update the primary image with delayed/deferred reboot and show that the image is updated while the prior version is still the running one. Upon reboot, the updated primary image is active and in effect.
- Demonstrated TOE ability to update the secondary image with delayed/deferred reboot and show that the image is updated while the prior version is still the running one. Upon reboot, the updated primary image is active and in effect.

- This satisfies the testing requirement.

FPT_TUD_EXT.1 Test #2

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.

- Test Steps:
  - TOE: Save copy OF configuration.

  - For Each of the FAILED UPDATE IMAGES
    1. TOE: show version of ACOS & confirm older version (for BOTH primary and secondary)
    2. HTTPS/SFTP/SCP Server download FAILED UPDATE IMAGE for update by ACOS

3. TOE: update ACOS from the failed update image – primary image
4. TOE: update ACOS from the failed update image – secondary image

5. Observe: all update operations FAIL and logged events indicating the reason for update failure (e.g. corrupt image, no signature, or bad signature).

- TOE: reboot – to primary image
- Observe: primary version is unchanged
- TOE: reboot – to secondary image
- Observe: secondary version is unchanged
- ACOS Record: CLI outputs and ACOS Logged Events
- TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated TOE's ability to show the current installed versions of ACOS, both for primary and secondary installed images, prior to attempting to install bad (e.g. corrupt image, no signature, or bad signature) images.
  - Demonstrated the TOE's ability to detect updated version of images with invalid signature and fail the trusted update operation, for both primary and secondary images.
  - Demonstrated the TOE's ability to detect updated version of images with no signature and fail the trusted update operation, for both primary and secondary images.
  - Demonstrated the TOE's ability to detect updated version of images with corrupted/modified content and fail the trusted update operation, for both primary and secondary images.
  - Demonstrate dthat the TOE's installed image versions and current running version are unchanged after these failed update operations.
  - Demonstrated that failed attempts to update the TOE do not affect the running version or image versions installed on the TOE.
- This satisfies the testing requirement.

FPT_TUD_EXT.1 Test #3

~~**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 (ifapplicable) for all TOE components.~~

- N/A – Trusted updates by published hash are not claimed.
  - The TOE is not a distributed TOE

2.2.5.5 FPT_STM_EXT.1 Reliable Time Stamps

FPT_STM_EXT.1

The evaluator shall **perform the following tests**:

FPT_STM_EXT.1 Test #1

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.

- Test Steps:
  - TOE: Save copy OF configuration.
  - TOE: Show and record the current system time (`show clock`)
  - TOE: Set a new system time (`clock set`)
    (ensure that date is different than the current date on the DUT)
  - TOE: Show the system time (`show clock`)
  - Observe: System time changed to the new setting
  - ACOS Record: CLI outputs and ACOS Logged Events
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that security administrator can set the local clock and that the new time setting is reflected in subsequent event logs on the TOE
  - Demonstrated that time change is logged indicating both the new and prior system times.
- This satisfies the testing requirement.

FPT_STM_EXT.1 Test #2

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.~~

- Test Steps:
  - TOE: Save copy OF configuration.
  - TOE: Set a new system time (`clock set`)
    (ensure that date/time is different than that used by the NTP Servers)
  - TOE: Show and record current system time (`show clock`)
  - NTP Server #1: Ensure that the NTP server time is notably different than the TOE
  - TOE: Configure for NTP Server #1 **(w/ connectivity via IPsec VPN)**
  - TOE: “enable” synchronization with NTP Server #1
  - Observe: NTP Status (`show ntp status`) for the NTP Servers
  - Observe: NTP Servers #1 indicates “synchronized” status.

- Observe: ACOS system time is consistent with the NTP Server #1 network time

- Result:
- **Verdict: Pass**
  - Demonstrated that the TOE can be configured to synchronize system time with an external NTP server via IPsec tunnel and that it does affect the system time of the TOE as reflected in subsequent ACOS audit and/or event log entries.

  - Demonstrated that the TOE is using NTP version 4.

- This satisfies the testing requirement.

4.2.6 TOE Access (FTA)

FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1

The evaluator shall **perform the following test**:

FTA_SSL_EXT.1 Test #1

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

- Test Steps:
  - TOE: Save copy OF configuration.
  - TOE Configure ACOS for a terminal idle timeout of 3 minutes (`terminal idle-timeout 3`)
  - ACOS Console: login for user “niap-user1”
  - WAIT for 3 minutes with no activity on the console session
  - Observe: Console session has been terminated & requires a new login
  - Observe: Termination of session event(s) logged with an idle timeout reason
  - TOE Configure ACOS for a terminal idle timeout of 5 minutes (`terminal idle-timeout 5`)
  - ACOS Console: login for user “niap-user1”
  - WAIT for 5 minutes with no activity on the console session
  - Observe: Console session has been terminated & requires a new login

    - Observe: Termination of session event(s) logged with an idle timeout reason
    - Record: CLI outputs and ACOS Logged Events
    - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - Demonstrated that the TOE can be configured for different periods of inactivity (aka idle-time) after which successfully authenticated (logged in) interactive management sessions to the CLI of the **TOE console** will be terminated, requiring subsequent authentication to perform subsequent management operations on the TOE.
    - Demonstrated that the TOE terminates such sessions after the configured inactivity (aka idle) time as indicated in the logs.

- This satisfies the testing requirement.

FTA_SSL.3 TSF-initiated Termination (Refinement)

FTA_SSL.3

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

FTA_SSL.3 Test #1a

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 Steps:
    - TOE: Save copy OF configuration.
    - TOE Configure ACOS for a terminal idle timeout of 3 minutes (`terminal idle-timeout 3`)
    - Linux Client: login via SSH for user “niap-user1”
    - WAIT for 3 minutes with no activity on the SSH session
    - Observe: SSH session has been terminated & requires a new connection and/or login
    - Observe: Termination of session event(s) logged with an idle timeout reason
    - TOE Configure ACOS for a terminal idle timeout of 5 minutes (`terminal idle-timeout 5`)
    - Linux Client: login via SSH for user “niap-user1”
    - WAIT for 5 minutes with no activity on the SSH session
    - Observe: SSH session has been terminated & requires a new connection and/or login
    - Observe: Termination of session event(s) logged with an idle timeout reason
    - Record: CLI outputs and ACOS Logged Events
    - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that the TOE can be configured for different periods of inactivity (aka idle-time) after which successfully authenticated (logged in) interactive management sessions remotely to the **CLI using SSH via IPsec trusted path** will be terminated, requiring subsequent authentication to perform subsequent management operations on the TOE.
  - Demonstrated that the TOE terminates such sessions after the configured inactivity (aka idle) time as indicated in the logs.

- This satisfies the testing requirement.

FTA_SSL.3 Test #1b

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.

- Procedure:
  - TOE: Save copy OF configuration.

  - TOE: Configure ACOS for a GUI idle timeout of 3 minutes (`web-service gui-timeout-policy idle 3`)
  - GUI Client: login via GUI for user "niap-user1"
  - WAIT for 3 minutes with no activity on the GUI session
  - Observe: GUI session has been terminated & requires a new login
  - Observe: Termination of session event(s) logged with an idle timeout reason

  - TOE: Configure ACOS for a GUI idle timeout of 5 minutes (`web-service gui-timeout-policy idle 5`)
  - GUI Client: login via GUI for user "niap-user1"
  - WAIT for 5 minutes with no activity on the GUI session
  - Observe: GUI session has been terminated & requires a new login
  - Observe: Termination of session event(s) logged with an idle timeout reason

  - Record: CLI outputs and ACOS Logged Events
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that the TOE can be configured for different periods of inactivity (aka idle-time) after which successfully authenticated (logged in) interactive management sessions to the **GUI via IPsec of the TOE** will be terminated, requiring subsequent authentication to perform subsequent management operations on the TOE.

    - Demonstrated that the TOE terminates such sessions after the configured inactivity (aka idle) time as indicated in the logs.

- This satisfies the testing requirement.

FTA_SSL.4 User-initiated Termination (Refinement)

FTA_SSL.4

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

FTA_SSL.4 Test #1

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.

- Test Steps:
    - TOE Console: login for user "niap-user1"
    - TOE Console: enter privileged EXEC mode (`enable`), enter configuration level (`configure`)
    - Observe: login, priv-EXEC mode entry, and CONFIG-level entry events are logged
    - TOE Console: exit configuration level (`exit`)
    - Observe: CONFIG-level exit event are logged
    - TOE Console: exit priv-EXEC mode (`exit`)
    - Observe: priv-EXEC exit event are logged
    - TOE Console: logout (`exit`)
    - Observe: logout event logged
    - Record: CLI outputs and ACOS Logged Events

- Result:
- **Verdict: Pass**
    - Demonstrated that interactive management sessions to the CLI of the TOE console can be terminated by the administrative user using the 'exit' CLI command.

        - This satisfies the testing requirement.

FTA_SSL.4 Test #2a

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 Steps:

- Linux Client: Start collection of PCAP for TCP/22
  - Linux Client: login via SSH for user “niap-user1”
  - Linux Client: enter privileged EXEC mode (`enable`), enter configuration level (`configure`)
  - Observe: login, priv-EXEC mode entry, and CONFIG-level entry events are logged
  - Linux Client: exit configuration level (`exit`)
  - Observe: CONFIG-level exit event are logged
  - Linux Client: exit priv-EXEC mode (`exit`)
  - Observe: priv-EXEC exit event are logged
  - Linux Client: logout (`exit`)
  - Observe: logout event logged

  - Record: CLI outputs and ACOS Logged Events

- Result:
- **Verdict: Pass**
  - Demonstrated that interactive management sessions remotely to the CLI using SSH trusted path (SSH via IPsec) can be terminated by the administrative user using the ‘exit’ CLI command.

- This satisfies the testing requirement.

FTA_SSL.4 Test #2b

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 Steps:
  - GUI Client: Start collection of PCAP for TCP/443
  - GUI Client: login via GUI for user “niap-user1”
  - Observe: login events are logged
  - GUI Client: click the LOGOUT button
  - Observe: login session exits (e.g. logout) event are logged
  - Record: CLI outputs and ACOS Logged Events

- Result:
- **Verdict: Pass**
  - Demonstrated that interactive management sessions remotely to the GUI trusted path (GUI via IPsec) can be terminated by the administrative user using the ‘exit’ CLI command.

- This satisfies the testing requirement.

FTA_TAB.1 Default TOE Access Banners (Refinement)

FTA_TAB.1

The evaluator shall also perform **the following test**:

FTA_TAB.1 Test #1

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 Steps:
  - TOE: Save copy OF configuration.
  - TOE: Make sure ACOS is enabled for FIPS mode (show version should indicate FIPS mode)
  - TOE: Configure for pre-login SSH/console banner (CLI: `banner login`) saying "SSH/CONSOLE – Test Pre-Login Message"
  - TOE: Configure for pre-login SSH banner (CLI: `banner exec`) saying "SSH/CONSOLE – Test Post-Login Message"
  - SSH Client: Login to TOE
    1. Observe "SSH/CONSOLE – Test Pre-Login Message" configured banner is displayed PRIOR TO prompting for Password.
    2. Observe "SSH/CONSOLE – Test Post-Login Message" configured banner is displayed AFTER successful login.
  - TOE Console: Login to TOE
    1. Observe "SSH/CONSOLE – Test Pre-Login Message" configured banner is displayed PRIOR TO prompting for Password.
    2. Observe "SSH/CONSOLE – Test Post-Login Message" configured banner is displayed AFTER successful login.
  - TOE: Configure for pre-login GUI banner (GUI @ /gui/#/system/settings/web "Pre GUI Login Message) saying "GUI – Test Pre-Login Message"
  - TOE: Configure for post-login GUI banner (GUI @ /gui/#/system/settings/web) "GUI Login Message saying "GUI – Test Post-Login Message"
  - GUI Client: Login to TOE
    1. Observe "GUI – Test Pre-Login Message" configured banner is displayed PRIOR TO prompting for Username/Password.
    2. Observe "GUI – Test Post-Login Message" configured banner is displayed AFTER successful login.
  - ACOS Record: CLI outputs and ACOS Logged Events
  - ACOS Record: GUI display of login pages including the pre-login and post-login banner messages.
  - TOE: Restore configuration to saved settings.
- Result:
- **Verdict: Pass**

- Demonstrated ability to configure separate banner messages for CLI access to the TOE to be displayed BEFORE authentication occurs and AFTER successful authentication occurs.

- Demonstrated that when accessing using CLI remotely using SSH (via IPsec)
  1. the “CLI login banner” is displayed PRIOR to authentication parameters being prompted.
  2. the “CLI exec banner” is displayed AFTER to a successful authentication.

- Demonstrated that when accessing using CLI on the local TOE console
  1. the “CLI login banner” is displayed PRIOR to authentication parameters being prompted.
  2. the “CLI exec banner” is displayed AFTER to a successful authentication.

- Demonstrated ability to configure separate banner messages for GUI access to the TOE to be displayed BEFORE authentication occurs and AFTER successful authentication occurs.

- Demonstrated that when accessing using GUI remotely (via IPsec)
  1. the “GUI login banner” is displayed PRIOR to authentication parameters being prompted.
  2. the “GUI exec banner” is displayed AFTER to a successful authentication.

- This satisfies the testing requirement.

4.2.7 Trusted path/channels (FTP)

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1

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**:

FTP_ITC.1 Test #1a2 – Syslog

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.

- Test Steps:
  - TOE: Save copy OF configuration.
  - Linux Server: Configure test server (as described above)
  - Linux Server: Start Syslog server/service.

  - Linux Server: Start collection of PCAP for UDP/514
  - TOE: Configure ACOS for Syslog (see above)
  - TOE: Perform operations that generate SYSLOG entries (e.g. file import operations using SCP)
  - Observe: Traffic on PCAP @ Linux Server

  - The following steps are not applicable since SYSLOG is operating over UDP.
    - ~~Switch #1 Interrupt traffic for 15 seconds from TOE to Linux Server~~
      ~~(a switch rule to drop traffic to Linux Server could work here)~~
    - ~~Switch #1 Restore traffic for TOE to Linux Server~~

    - ~~Observe: Traffic on PCAP @ Linux Server~~
  - Switch #1 Interrupt traffic for 5 minutes from TOE to Linux Server
    (a switch rule to drop traffic to Linux Server could work here)
  - Observe: Connection to Syslog server LOST
  - Switch #1 Restore traffic for TOE to Linux Server
  - Observe: ACOS re-connects to Syslog server
  - Record: How long it takes for ACOS to reestablish connection or if or login/CLI-commands prompted ACOS to re-connect
  - Record: PCAP of exchanges from TOE and Linux Server
  - ACOS Record: CLI outputs and ACOS Logged Events
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated Trusted Channel set-up for SYSLOG over IPsec from the TOE to the Trusted Syslog server
  - Demonstrated Trusted Channel communications for SYSLOG over IPsec are initiated by the TOE.

  - Demonstrated behavior of the SYSLOG over IPsec trusted channel during communication outages between the TOE and the SYSLOG server with:
    1. Log entries immediately (or at next DUT audit/event log instance) are conveyed to the SYSLOG server at the end of an outage

  - Demonstrated that when SYSLOG over IPsec communication outages end, subsequent exchanges with the trusted channel's server are secured over IPsec.
  - This satisfies the testing requirement.

FTP_ITC.1 Test #1l – NTP

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.

- Test Steps:

- TOE: Save copy OF configuration.
- Linux Server: Configure test server (as described above)
- Linux Server: Start NTP server/service.
- Linux Server: Start collection of PCAP for UDP/123
- TOE: show ACOS system time
- Observe: ACOS system time is from local device clock settings
- TOE Configure ACOS for NTP (see above)
- TOE: "enable" synchronization with NTP Server #1 (`ntp server 10.65.25.67 enable`)
- Observe: Traffic on PCAP @ Linux Server
- Observe: NTP Status (`show ntp status`) for the NTP Servers
- Observe: NTP Servers #1 indicates "synchronized" status.
- Observe: ACOS system time is consistent with the NTP Server #1 network time
- The following steps are not applicable since NTP is operating over UDP.
  - ~~Switch #2 Interrupt traffic for 15 seconds from TOE to Linux Server (a switch rule to drop traffic to Linux Server could work here)~~
  - ~~Switch #2 Restore traffic for TOE to Linux Server~~
  - ~~Observe: Traffic on PCAP @ Linux Server~~
- Switch #2 Interrupt traffic for 5 minutes from TOE to Linux Server (a switch rule to drop traffic to Linux Server could work here)
- Observe: NTP Status (`show ntp status`) for the NTP Servers
- Observe: NTP Servers #1 indicates "polling" status.
- Switch #1 Restore traffic for TOE to Linux Server
- Observe: ACOS re-connects to NTP server
- Observe: NTP Status (`show ntp status`) for the NTP Servers
- Observe: NTP Servers #1 indicates "synchronized" status.
- Record: How long it takes for ACOS to reestablish time synchronization
- Record: PCAP of exchanges from TOE and Linux Server
- ACOS Record: CLI outputs and ACOS Logged Events
- TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated Trusted Channel set-up for NTP over IPsec from the TOE to the Trusted NTP server
  - Demonstrated Trusted Channel communications for NTP over IPsec are initiated by the TOE.
  - Demonstrated behavior of the NTP over IPsec trusted channel during communication outages between the TOE and the NTP server with
    1. Communication outage after NTP server configuration and prior to or during polling state is considered part of polling activities and not connection outages. Accordingly, nothing is logged in these conditions.
    2. ~5 – 45 minutes to fully resynchronize the TOE with NTP server after outage is ended.

    - Demonstrated that when NTP over IPsec communication outages end, subsequent exchanges with the trusted channel's server are secured over IPsec.
- This satisfies the testing requirement.

FTP_ITC.1 Test #1o1 – SSH SCP/SFTP - password

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.

- Test Steps:
    - TOE: Save copy OF configuration.
    - Linux Server: Configure test server (as described above)
    - Linux Server: Set up and start (if needed) SCP and SFTP services.
    - Linux Server: Start collection of PCAP for TCP/22
    - SSH Client #1: Initiate a new SSH session for "admin-pw-rw-user1" w/ password "user1-123456"
    - TOE: Import something innocuous (link an AFLEX file) with SCP
    - TOE: Export something innocuous (link an AFLEX file) with SCP
    - Observe: Traffic on PCAP @ Linux Server
    - Observe: File transfer operations SUCCEEDS
    - TOE: Import something innocuous (link an AFLEX file) with SFTP
    - TOE: Export something innocuous (link an AFLEX file) with SFTP
    - Observe: File transfer operations SUCCEEDS
    - Switch #1 Interrupt traffic for 5 minutes from TOE to Linux Server
      (a switch rule to drop traffic to Linux Server could work here)
    - TOE: REPEAT IMPORT/EXPORT w/ SCP ABOVE
    - TOE: REPEAT IMPORT/EXPORT w/ SFTP ABOVE
    - Observe: File transfer operations FAIL (including logged reason for failure)
    - Switch #1 Restore traffic for TOE to Linux Server
    - TOE: REPEAT IMPORT/EXPORT w/ SCP ABOVE
    - TOE: REPEAT IMPORT/EXPORT w/ SFTP ABOVE
    - Observe: File transfer operations SUCCEED
    - TOE: REPEAT IMPORT/EXPORT w/ SCP ABOVE with bad remote filename
    - TOE: REPEAT IMPORT/EXPORT w/ SFTP ABOVE with bad remote filename
    - Observe: Both import operations FAIL (including logged reason for failure)
    - TOE: REPEAT IMPORT/EXPORT w/ SCP ABOVE with bad user name
    - TOE: REPEAT IMPORT/EXPORT w/ SFTP ABOVE with bad user name
    - Observe: Both import operations FAIL (including logged reason for failure)
    - TOE: REPEAT IMPORT/EXPORT w/ SCP ABOVE with bad password
    - TOE: REPEAT IMPORT/EXPORT w/ SFTP ABOVE with bad password
    - Observe: Both import operations FAIL (including logged reason for failure)
    - TOE: REPEAT IMPORT/EXPORT w/ SCP ABOVE with bad IP address of Server

- TOE: REPEAT IMPORT/EXPORT w/ SFTP ABOVE with bad IP address of Server
- Observe: Both import operations FAIL (including logged reason for failure)
- Record: PCAP of exchanges from TOE and Linux Server
- ACOS Record: CLI outputs and ACOS Logged Events
- TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated Trusted Channel set-up for SSH SCP/SFTP over IPsec from the TOE to the Trusted SCP/SFTP File server
  - Demonstrated Trusted Channel communications for SSH SCP/SFTP over IPsec are initiated by the TOE.
  - Demonstrated behavior of the SSH SCP/SFTP over IPsec trusted channel during communication outages between the TOE and the SYSLOG server
  - Demonstrated that when SSH SCP/SFTP over IPsec communication outages end, subsequent exchanges with the trusted channel's server are secured over IPsec and transact successfully
- This satisfies the testing requirement.

FTP_ITC.1 Test #2 – TOE Initiated Trusted Channels

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.

NOTE: Included in FTP_ITC.1 Test #1 test group for trusted channels initiated by the TOE.

1. Syslog
2. NTP
3. SSH SCP/SFTP

FTP_ITC.1 Test #3a – Trusted channels not in plain text – Syslog, SCP

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

- Test Steps:
  - TOE: Save copy OF configuration.
  - Linux Server: Configure test server (as described above)
  - Linux Server: Start Syslog server/service (if not running)
  - Linux Server: Start SCP/SFTP server/service (if not running)
  - Linux Server: Start collection of PCAP for UDP/514, TCP/22 or all packets from DUT,
  - IPSec VPN GW: Start packet logging and capture of encrypted packets link to the DUT
  - TOE: Configure ACOS for Syslog (see above)
  - TOE: Configure ACOS for SCP (see above)

  - TOE: Perform file import operations using SCP
  - Observe: Traffic on PCAP @ Linux Server
  - Observe: Syslog entries on the server
  - Observe: SCP transfers succeed
  - IPSec VPN GW: Take down the tunnel
  - Observe: IPsec tunnel marked as down on the DUT
  - Observe: Syslog entries on the server cease
  - Observe: SCP transfers fail
  - IPSec VPN GW: Bring the tunnel back up
  - Observe: IPsec tunnel reestablishes and is marked as up on the DUT (incl how long it took to do so)
  - Observe: Syslog entries on the server resume
  - Observe: SCP transfers again succeed
  - Record: How long it takes for ACOS to reestablish connection or if or login/CLI-commands prompted ACOS to re-connect
  - Record: PCAP of exchanges observed at the the Linux Server
  - Record: PCAP of exchanges observed at the IPsec VPN GW's link to DUT
  - Record: Packets log on the IPsec VPN GW (showing both IPsec encrypted to/from DUT & cleartext @ link to Linux server.
  - ACOS Record: CLI outputs and ACOS Logged Events
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated Trusted Channel set-up for SYSLOG over IPsec from the TOE to the Trusted Syslog server
  - Demonstrated Trusted Channel communications for SSH SCP/SFTP over IPsec and SYSLOG over IPsec are initiated by the TOE.
  - Demonstrated Trusted Channels for SSH SCP/SFTP over IPsec and SYSLOG over IPsec from the TOE to the trusted channel peers are not in plain text, as they are encrypted over IPSec.
  - Demonstrated Trusted Channels for SSH SCP/SFTP over IPsec and SYSLOG over IPsec at the point of ingress/egress from the TOE are not in plain text
  - Demonstrated failure of IPsec tunnel due to communication outage with IPsec test peer and related detection and local logging.
  - Demonstrated IPsec tunnel is reestablished when communication outage with IPsec test peer is over, allowing Trusted Channels communications SYSLOG over IPsec and SSH SCP/SFTP over IPsec via the tunnel is restored and continues to not be in plain-text (e.g. secured via IPsec).
    1. SYSLOG entries are immediately (or at next DUT audit/event log instance) conveyed to the SYSLOG server and logged at the end of an outage.
    2. SCP/SFTP operations will immediately (or at the next Security Admin operation) begin to succeed at the end of an outage.

- This satisfies the testing requirement.

FTP_ITC.1 Test #3b – Trusted channels not in plain text - NTP

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

- Test Steps:
    - TOE: Save copy OF configuration.
    - Linux Server: Configure test server (as described above)
    - Linux Server: Start NTP server/service (if not running)
    - Linux Server: Start collection of PCAP for UDP/123 or all packets from DUT,

    - IPSec VPN GW: Start packet logging and capture of encrypted packets link to the DUT
    - TOE: Configure ACOS for NTP (see above)
    - TOE: Perform file import operations using SCP and SFTP
    - Observe: Traffic on PCAP @ Linux Server
    - Observe: NTP Synchronized to server
    - IPSec VPN GW: Take down the tunnel
    - Observe: IPsec tunnel marked as down on the DUT
    - Observe: NTP loses synchronization with server
    - IPSec VPN GW: Bring the tunnel back up
    - Observe: IPsec tunnel reestablishes and is marked as up on the DUT (incl how long it took to do so)
    - Observe: NTP regains synchronization with server (incl how long it took to do so)
    - Record: How long it takes for ACOS to reestablish connection or if or login/CLI-commands prompted ACOS to re-connect
    - Record: PCAP of exchanges observed at the the Linux Server
    - Record: PCAP of exchanges observed at the IPsec VPN GW's link to DUT
    - Record: Packets log on the IPsec VPN GW (showing both IPsec encrypted to/from DUT & cleartext @ link to Linux server.
    - ACOS Record: CLI outputs and ACOS Logged Events
    - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
    - Demonstrated Trusted Channel set-up for NTP over IPsec from the TOE to the Trusted Syslog server
    - Demonstrated Trusted Channel communications for NTP over IPsec are initiated by the TOE.
    - Demonstrated Trusted Channel for NTP over IPsec from the TOE to the trusted channel peers are not in plain text, as they are encrypted over IPSec.
    - Demonstrated Trusted Channel for NTP over IPsec at the point of ingress/egress from the TOE are not in plain text

  - Demonstrated failure of IPsec tunnel due to communication outage with IPsec test peer and related detection and local logging.
  - Demonstrated behavior of the NTP over IPsec trusted channel during communication outages between the TOE and the NTP server with
    1. < ~10 minutes (often 1 minute) to detect comm-to-NTP server lost during outage (and enter polling state)
    2. ~5 – 45 minutes to fully resynchronize the TOE with NTP server after outage is ended.
  - Demonstrated IPsec tunnel is reestablished when communication outage with IPsec test peer is over, allowing Trusted Channels communications NTP over IPsec via the tunnel is restored and continues to not be in plain-text (e.g. secured via IPsec).

- This satisfies the testing requirement.

FTP_ITC.1 Test #3c – Trusted channels not in plain text – IPsec Remotely Initiated/terminated

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

- Test Steps:
  - TOE: Save copy OF configuration.
  - IPSec VPN GW: Disable tunnel to DUT
  - TOE: Remove VPN GW and IPsec configuration items
  - TOE: Remove IPsec logging and cipher checking configuration items
  - TOE: Disable logging to the DUT console
  - TOE: Enable IPsec logging to ACOS event log
  - TOE: Enable IPsec cipher checking
  - TOE: Configure VPN GW settings per Guidance Document (with RSA or ECDSA Certificate/Key) – incl settings for defaults
  - TOE: Configure VPN IPsec settings per Guidance Document (with 1 tunnel to all servers or independent tunnels to each server) – incl settings for defaults
  - ACOS Record: CLI outputs and ACOS Logged Audits/Events
  - IPSec VPN GW: Enable Tunnel to DUT (RSA/ECDSA-based)
  - Observe: VPN GW initiating IPsec Tunnel (e.g. sending IKE_SA_INIT) to DUT
  - IPSec VPN GW: Disable Tunnel to DUT (RSA/ECDSA-based)
  - Observe: VPN GW terminating IPsec Tunnel (e.g. sending DELETE) to DUT
  - ACOS Record: CLI outputs and ACOS Logged Audits/Events
  - TOE: Modify VPN GW settings with Pre-shared Key (e.g. remove RSA/ECDSA & cert/key settings)
  - ACOS Record: CLI outputs and ACOS Logged Audits/Events
  - IPSec VPN GW: Update configuration (or support parallel configuration) to support Pre-shared key for authentication
  - IPSec VPN GW: Enable Tunnel to DUT (Pre-shaed key-based)
  - Observe: VPN GW initiating IPsec Tunnel (e.g. sending IKE_SA_INIT) to DUT

  - IPSec VPN GW: Disable Tunnel to DUT (Pre-shaed key-based)
  - Observe: VPN GW terminating IPsec Tunnel (e.g. sending DELETE) to DUT
  - ACOS Record: CLI outputs and ACOS Logged Audits/Events
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that IPsec Tunnels can be established and terminated from IPsec Peer to the TOE

- This satisfies the testing requirement.

FTP_ITC.1 Test #4 – Trusted channels connection outages

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 Steps:
  - NO STEPS: No separate testing needed as connection disruptions are included in FTP_ITC.1 Test #1 tests procedures.
  - The TOE is not a distributed TOE

FTP_TRP.1/Admin Trusted Path

FTP_TRP.1/Admin

The evaluator shall **perform the following tests**:

FTP_TRP.1/Admin Test #1a – SSH CLI, HTTPS/GUI via IPsec

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.

- Test Steps:
  - NO STEPS: Otherwise tested in FTA_SSL.4 Test #2a, FTA_SSL.4 Test #2b, and in FTP_TRP.1/Admin Test #2a and #2b.

FTP_TRP.1/Admin Test #2a – Trusted paths not in plain text – Test a

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

~~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 Steps:
  - TOE: Save copy OF configuration.
  - Linux Server: Configure test server (as described above)
  - Linux Server: Start Syslog server/service (if not running)
  - Linux Server: Start SCP/SFTP server/service (if not running)
  - TOE: Configure ACOS for Syslog (see above)
  - TOE: Configure ACOS for SCP (see above)
  - TOE: Perform file import operations using SCP
  - Observe: Traffic on PCAP @ Linux Server
  - Observe: Syslog entries on the server
  - Observe: SCP transfers succeed
  - Linux Server: Start collection of PCAP for TCP/443, TCP/22 to the DUT,
  - IPSec VPN GW: Start packet logging and capture of encrypted packets link to the DUT
  - Linux Server: Attempt SSH remote terminal session to DUT via IPsec tunnel and exit session
  - Linux Server: Attempt GUI HTTPS session to DUT via IPsec tunnel and exit session
  - Observe: SSH session connects successfully.
  - Observe: GUI session connects successfully

- Record: PCAP of exchanges observed at the the Linux Server
  - Record: PCAP of exchanges observed at the IPsec VPN GW's link to DUT
  - Record: Packets log on the IPsec VPN GW (showing both IPsec encrypted to/from DUT & cleartext @ link to Linux server).
  - ACOS Record: CLI outputs and ACOS Logged Events
  - TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated that all trusted paths (SSH-CLI and GUI-HTTPS) are encrypted via IPsec for their traffic to/from the TOE management port.
    1. The PCAP of IPSec VPN GW's link to the DUT confirm all traffic is encrypted.
    2. The PCAP on the Linux server confirms the traffic being tunneled is that of the SSH CLI and GUI-HTTPS administrator interface to the DUT.
    3. The packets log on the IPsec VPN GW correlates the IPsec encrypted packets with those exchanged by the VPN GW and the Linux server.

- This satisfies the testing requirement.

o The TOE is not a distributed TOE

FTP_TRP.1/Admin Test #2b – Trusted paths not in plain text – Test b

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

- Test Steps:
  - TOE: Save copy OF configuration.
  - Linux Server: Configure test server (as described above)
  - Linux Server: Start Syslog server/service (if not running)
  - Linux Server: Start SCP/SFTP server/service (if not running)
  - TOE: Configure ACOS for Syslog on Linux server (see above)
  - TOE: Configure ACOS for NTP Linux server (see above)
  - TOE: Configure ACOS for SCP Linux server (see above)
  - TOE: Perform file import operations using SCP and SFTP
  - Observe: Traffic traversing IPsec tunnel to Linux Server
  - Observe: Syslog entries on the server
  - Observe: NTP synchronized with Linux server
  - Observe: SCP transfers succeed
  - Linux Server: Attempt SSH remote terminal session to DUT via IPsec tunnel and exit session
  - Linux Server: Attempt GUI HTTPS session to DUT via IPsec tunnel and exit session
  - Observe: SSH session connects successfully.
  - Observe: GUI session connects successfully

- Linux Client: Start collection of PCAP for all packets with DUT,
- Linux Client: Attempt SSH remote terminal session to DUT not via IPsec tunnel
- Linux Client: Attempt GUI HTTP session to DUT not via IPsec tunnel
- Observe: SSH and GUI sessions does not connect.
- TOE: Add BYPASS ACLs to permit SSH and GUI from Linux Client
- Linux Client: Attempt SSH remote terminal session to DUT not via IPsec tunnel
- Linux Client: Attempt GUI HTTP session to DUT not via IPsec tunnel
- Observe: SSH and GUI sessions successfully establish with native SSH and HTTPS exchanges from the Linux Client not protected (e.g. BYPASSING) the IPsec tunnel.

- Record: PCAP of exchanges observed at the the Linux Client.

- ACOS Record: CLI outputs and ACOS Logged Events
- TOE: Restore configuration to saved settings.

- Result:
- **Verdict: Pass**
  - Demonstrated the ability to configure the TOE to BYPASS IPsec encryption to communicate with trusted channel client (SSH-CLI, GUI-HTTPS) using ACLs
    1. The PCAP of Linux Client to the DUT confirmed all traffic is as natively generated (effective cleartext) from the test client and is not IPsec encrypted.
       - Even though SSH and HTTPS/TLS are encrypting these trusted channels, SSHS and TLSS SFRs are not claimed in the Security Target and hence this traffic is considered effectively clear text for this evaluation.
- This satisfies the testing requirement.

5. SAR ASSURANCE ACTIVITIES

5.1 Life Cycle Support (ALC)

ALC_CMC.1

Verdict: Pass

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."

We have verified that the TOE provided for evaluation is labeled with its reference. The product is a hardware product, and the model of the product is printed on the product (on a label). The model number is printed on the left upper front cover of the product.

The model number is specified as follows:

TH-4435, TH-5840-11, TH-7445, TH-7650-11, TH-7655

The product provides a capability to output its model and its firmware version to the CLI and Web interfaces. The command used to output the model and the firmware version is "show version". Version information is also available through the web interface.

We have visually inspected the A10 Networks products and verified that the version was displayed appropriately. We also used the CLI command "show version" and verified that output of the command matches the versions that is described in the ST.

We have verified that the TOE references used are consistent. The model and software version numbers are consistent with information specified in the ST and CCCG.

ALC_CMS.1

Verdict: Pass

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."

We have determined that configuration list includes the following set of items:
a) the TOE itself;
b) the evaluation evidence required by the SARs.

ST Section 1.1 and Section 1.2.2 identify the TOE.

TOE Reference:	A10 Networks Thunder Series Appliances
TOE Hardware Models:	TH-4435, TH-5840-11, TH-7445, TH-7650-11, TH-7655
TOE Software Version:	ACOS 5.2.1-P3
TOE OS Kernel Version:	Linux 4.19 LTS

We have determined that all documents comprising the evaluation evidence are included in the configuration list.

We have determined that the configuration uniquely identifies each configuration item. A10 Networks Lifecycle Documentation describes unique identification of the configuration items. We have also used the output of the CM system provided by the vendor to verify unique identification of the configuration items used for development.

Component Verdict: Pass

5.2 Guidance Documents (AGD)

Operational user guidance (AGD_OPE.1)

Verdict: Pass

We performed the CEM work units associated with the AGD_OPE.1 SAR. Specific requirements and EAs on the guidance documentation were identified (where relevant) in the individual EAs for each SFR.

We ensured that the Operational guidance documentation is distributed to Security Administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that Security Administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

Security Target Section 1.2.2 “TOE Documentation” describes the TOE Operational guidance documentation.

A10 Networks CCCG documentation Section 1.2 “TOE Documentation” states that the documentation is downloadable from the A10 Networks support web site (https://support.a10networks.com).

We ensured that the Operational guidance is provided for the Operational Environment that the product supports as claimed in the Security Target and adequately address all platforms claimed for the TOE in the Security Target.

Section 1.3.2 “Elements Included in the TOE Operational Environment” in A10 Networks CCCG documentation describes elements included in the TOE Operational Environment.

Section 1.3.1 “TOE Models Evaluated” in A10 Networks CCCG documentation describes the TOE Models evaluated.

We ensured that the Operational guidance contains instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE.

Section 2.2 “Initial Setup via TOE Local Console” in A10 Networks CCCG documentation states that Cryptographic engine parameters are all preconfigured at the factory or are defaulted in the TOE. They should not be changed by the administrators of the TOE.

We ensured that the Operational guidance makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

Section 3 “Secure Management” in A10 Networks CCCG states that “This section describes TOE capabilities and provides further guidance on configuring and managing the TOE functionality when deployed in a CC evaluated configuration. It summarizes information in the ACOS Configuration Guides indicated in Section 1.2 above as pertains to the CC evaluated configuration for the TOE.

TOE administrators should review these ACOS Configuration Guides to establish a broad awareness of the range of functionality and capabilities in ACOS, including functions and capabilities that may be beyond the scope of the CC evaluated configuration.”

In addition we ensured that the following requirements are also met.

[TD0536] The documentation describes the process for verifying updates to the TOE for each method selected for FPT_TUD_EXT.1.3 in the Security Target. We verified that this process includes the following steps:

- Instructions for obtaining the update itself. This includes instructions for making the update accessible to the TOE (e.g., placement in a specific directory).

Section 3.9 “Trusted Updates” in A10 Networks CCCG documentation states that TOE update images are available from the A10 Networks support web site (https://support.a10networks.com). An A10 Networks support login ID and password is required to download these images.

- 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.

Section 3.9 “Trusted Updates” in A10 Networks CCCG documentation provides instructions for initiating the update process. The **upgrade** CLI command can be used. If the connection fails to establish or is lost while downloading the TOE update image from the trusted server, the update operation will be failed, and the administrator will have to retry the operation at a later time when reliable access to the server(s) is available. If any of the tests fail, the TOE update operation will be failed with visual feedback to the administrator initiating the update, along with corresponding logged failure events.

The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The guidance documentation makes it clear to an administrator which security functionality is covered by the Evaluation Activities.

Section 3 “Secure Management” in A10 Networks CCCG states that “This section describes TOE capabilities and provides further guidance on configuring and managing the TOE functionality when deployed in a CC evaluated configuration. It summarizes information in the ACOS Configuration Guides indicated in Section 1.2 above as pertains to the CC evaluated configuration for the TOE.

TOE administrators should review these ACOS Configuration Guides to establish a broad awareness of the range of functionality and capabilities in ACOS, including functions and capabilities that may be beyond the scope of the CC evaluated configuration.”

Preparative procedures (AGD_PRE.1)

Verdict: Pass

We performed the CEM work units associated with the AGD_PRE.1 SAR. Specific requirements and EAs on the preparative documentation were identified (and where relevant are captured in the Guidance Documentation portions of the EAs) in the individual EAs for each SFR.

We have examined the Preparative procedures and ensured they include a description of how the Security 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) in Section 1.3.4 of the CCCG.

We have examined the Preparative procedures and ensured they are provided for every Operational Environment that the product supports as claimed in the Security Target and adequately address all platforms claimed for the TOE in the Security Target.

Section 1.3.2 “Elements Included in the TOE Operational Environment” in A10 Networks CCCG documentation describes elements included in the TOE Operational Environment.

Section 1.3.4 “Operational Environment Assumptions” in A10 Networks CCCG documentation describes Operational Environment assumptions.

Section 1.3.1 “TOE Models Evaluated” in A10 Networks CCCG documentation describes the TOE Models evaluated.

Section 1.2 “TOE Documentation” references applicable installation documents addressing the 5 target models.

We have examined the preparative procedures to ensure they include instructions to successfully install the TSF in the Operational Environment.

We have examined 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.

Section 2.2.1 “Initial Connection and FIPS Mode Confirmation” in A10 Networks CCCG documentation describes initial connection.

Section 2.2.2 “Basic System Parameters Configuration” in A10 Networks CCCG documentation describes system parameters configuration.

Section 2.2.3 “Establish an IPsec Tunnel for Connections in the Operational Environment” in A10 Networks CCCG documentation describes establishment of an IPsec tunnel for connections in the Operational Environment.

Section 2.2.2“Basic System Parameters Configuration” in A10 Networks CCCG documentation describes change of the default password.

Section 3.6 “Failed Authentication Lockout” in A10 Networks CCCG documentation describes configuration of the lockout feature.

In addition, we ensured that the following requirements are also met.

The preparative procedures

a) include instructions to provide a protected administrative capability;

We ensured that the preparative procedures include instructions to provide a protected administrative capability and for changing the default administrator password. Sections 2.2, 3.9, and 3.11 were reviewed. The A10 Networks CCCG documentation describes changing the default password of the default administrator account (admin) and configuring IPsec for remote administration.

and

a) identified TOE passwords that have default values associated with them and instructions are provided for how these can be changed.

Section 2.2.2“Basic System Parameters Configuration” in A10 Networks CCCG documentation describes change of the default password of the “admin”.

A10 Networks CCCG Section 2.2.1 describes authentication using default credentials.

Component Verdict: **Pass**

5.3 Development (ADV)

Basic Functional Specification (ADV_FSP.1)

Verdict: **Pass**

The Guidance Section 1.3.4 states that the TOE is a closed-system and does not support general-purpose computing capabilities. Only appliance specific interfaces are supported; including CLI (local console and remote SSH), and GUI. For example, general OS Shell access is not available from the appliance

The Guidance Section 1.3.2 states that:

SSH Clients: The TOE can be managed via IPsec from terminal clients on remote administrator workstations accessing the management Command Line Interface (CLI) of the TOE.

Web GUI Browser Clients: The TOE can be managed via IPsec from web browsers on remote administrator workstations accessing the management web GUI of the TOE.

The Guidance Section 3.13 describes the parameters and configuration of the IPsec protocol.

Paragraph 561 from the CEM: “In the case where the developer has provided adequate documentation to perform the analysis called for by the rest of the work units for this component without explicitly identifying SFR-enforcing and SFR-supporting interfaces, ADV_FSP.1-4 work unit should be considered satisfied.” Since the rest of the ADV_FSP.1 work units have been satisfied upon completion of the EAs, it follows that this work unit is satisfied as well.

We examined the interface documentation to developed a mapping of the interfaces to SFRs.

EAs that are associated with the SFRs are performed to ensure that all the SFRs where the security functionality is externally visible (i.e. at the TSFI) are covered. Therefore, the intent of the ADV_FSP.1-6 work unit is covered.

EAs that are associated with the SFRs are performed to ensure that all the SFRs where the security functionality is externally visible (i.e. at the TSFI) are addressed, and that the description of the interfaces is accurate with respect to the specification captured in the SFRs. Therefore, the intent of the ADV_FSP.1-7 work unit is covered.

Component Verdict: **Pass**

5.4 Vulnerability Assessment (AVA)

Vulnerability Survey (AVA_VAN.1)

Verdict: **Pass**

5.6.1.1 Evaluation Activity (Documentation): 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.

TD0547: NIT Technical Decision for Clarification on developer disclosure of AVA_VAN

The developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components should identify at a minimum the processors used by the TOE. Software components include applications, the operating system and other major components that are independently identifiable and reusable (outside of the TOE), for example a web server, protocol or cryptographic libraries, (independently identifiable and reusable components are not limited to the list provided in the example). This additional documentation is merely a list of the name and version number of the components and will be used by the evaluators in formulating vulnerability hypotheses during their analysis.

5.6.1.2 Evaluation Activity:

The evaluator formulates hypotheses in accordance with process defined in Appendix A. The evaluator documents the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3. The evaluator shall perform vulnerability analysis in accordance with Appendix A.2. The results of the analysis shall be documented in the report according to Appendix A.3.

We performed the search on the sources listed below to determine a list of potential flaw hypotheses that are more recent that the publication date of the cPP. The search was performed on 1/10/2023.

The following sources were used for the public search for vulnerabilities:

https://web.nvd.nist.gov/view/vuln/search
http://cve.mitre.org/cve/
https://www.cvedetails.com/vulnerability-search.php
http://www.kb.cert.org/vuls/html/search
www.exploitsearch.net
www.securiteam.com
http://nessus.org/plugins/index.php?view=search
http://www.zerodayinitiative.com/advisories
https://www.exploit-db.com/
https://www.rapid7.com/db/vulnerabilities

The following search terms were used to perform public search for vulnerabilities: Application Delivery Controller, Carrier Grade NAT, A10 Networks, A10 Thunder, TH-4435, TH-5840-11, TH-7445, TH-7650-11, TH-7655, ACOS, IPsec , IKE, NTP v.4, Xeon E5-2680v2, Xeon E5-2695v4, Xeon Gold 6258R, TCP, CentOS 7-9.2009, Apache HTTPD 2.4.46, OpenSSL 1.0.2k, Strongswan 5.0.4, NTP 4.2.6p5 and OpenSSH 7.4p1.

We performed an NMAP scan to find ports that are open. Fuzz testing was performed.

Verdict: the TOE is not vulnerable to the relevant attacks or vulnerabilities

Component Verdict: Pass

5.5 Tests (ATE)

Independent Testing (ATE_IND.1)

Verdict: Pass

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.

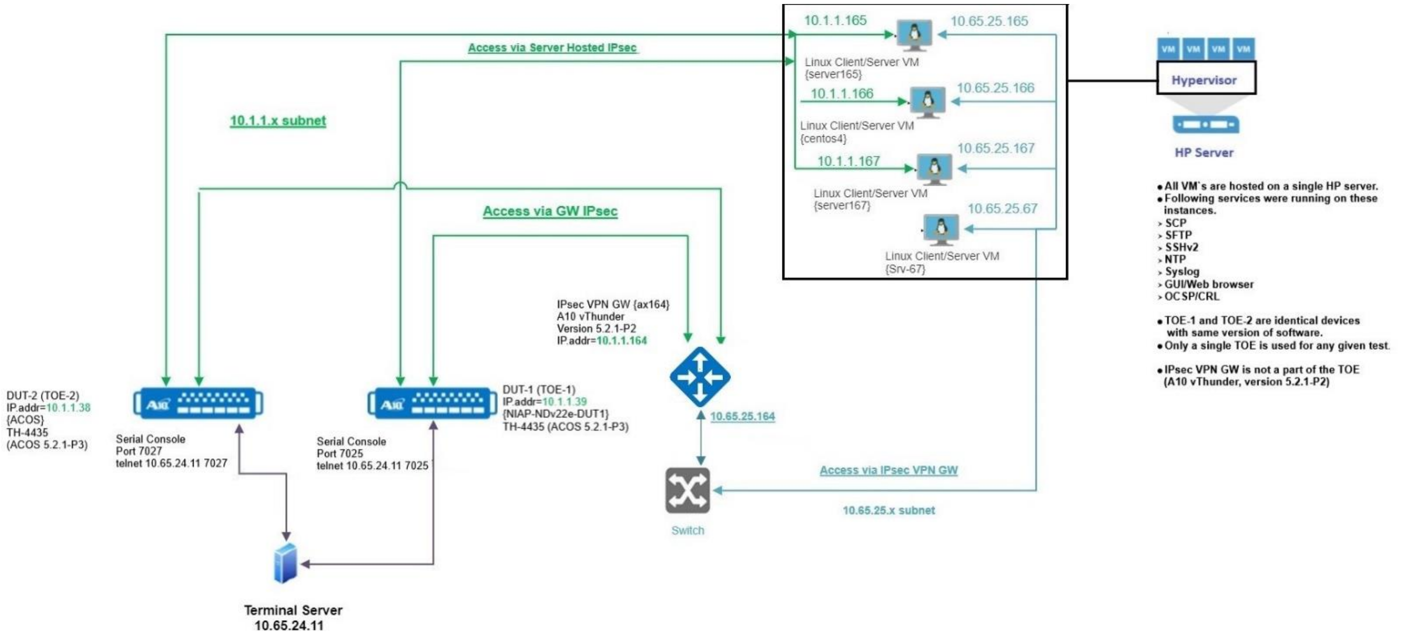
The evaluator performs the CEM work units associated with the ATE_IND.1 SAR. Specific testing requirements and EAs are captured for each SFR in Sections 2, 3 and 4.

The evaluator should consult Appendix 709 when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

The evaluator performed independent testing of the TOE as described in the SD and CEM. The evaluator created a Test Report. Based on the equivalency rationale, testing was performed on the following subset of TOE models: TH-4435.

Testbed Configuration and Tools

The following describes the configuration of the testbed used for testing activities and tools included in the testbed.



- All VM`s are hosted on a single HP server.
- Following services were running on these instances.
  - SCP
  - SFTP
  - SSHv2
  - NTP
  - Syslog
  - GUI/Web browser
  - OCSP/CRL
- TOE-1 and TOE-2 are identical devices with same version of software.
- Only a single TOE is used for any given test.
- IPsec VPN GW is not a part of the TOE (A10 vThunder, version 5.2.1-P2)

Figure 2: Test Configuration Diagram

The following provides information about the software packages/tools used in the test configuration.

-	SYSLOG (audit log) Server	syslog-ng 3 v3.30.1, v3.13.2
-	Nmap	7.91
-	SCP/SFTP Server	OpenSSH 7.6p1
-	CRL Distribution Point	OpenSSL 1.0.2k-fips, Apache HTTPD 2.4.6
-	OCSP Server	OpenSSL 1.0.2k-fips
-	NTP Server	ntpd 4.2.6p5, 4.2.6p3
-	SSH Client	OpenSSH 7.6p1
-	Web/GUI Browser	Firefox, 78.4.1esr(64-bit)
-	IPsec Peer (Server Hosted)	Linux strongSwan U5.6.2/K4.15.0-136-generic
-	IPsec Peer (VPN Gateway)	A10 ACOS vThunder 5.2.1-P2

Additional tools were used in testing for this configuration. These tools include the following.

-	Tcpdump	4.9.3, 4.9.2, 4.1.1
-	PuTTY	0.94
-	Nmap	7.91

Component Verdict: Pass

6. CONCLUSIONS

Overall Verdict: Pass

7. REFERENCES

IDENTIFIER	REFERENCE
[CC_PART1]	Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated April 2017, version 3.1, Revision 5
[CC_PART2]	Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated April 2017, version 3.1, Revision 5
[CC_PART3]	Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, April 2017, version 3.1, Revision 5
[CEM]	Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, April 2017, version 3.1, Revision 5
[NDcPPv2.2e]	collaborative Protection Profile for Network Devices, Version 2.2e, 23-March-2020
[ND-SD2.2]	Evaluation Activities for Network Device cPP, Version 2.2, December-2019
[ST]	A10 Networks Thunder Series Appliances TH-4435, TH-5840-11, TH-7445, TH-7650-11, TH-7655 with ACOS 5.2.1-P3 Security Target, Revision: 1.0
[CCCG]	A10 Networks Thunder Series Appliances TH-4435, TH-5840-11, TH-7445, TH-7650-11, TH-7655 with ACOS 5.2.1-P3 Common Criteria Configuration Guide, Revision: 1.0
[800-38A]	NIST Special Publication 800-38A Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques, December 2001
[800-56A Rev 3]	NIST Special Publication 800-56A, Revision 3, April 18, 2018 Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)
[FIPS 140-2]	FIPS PUB 140-2 Federal Information Processing Standards Publication Security Requirements for Cryptographic Modules, May 25, 2001
[FIPS PUB 186-4]	FIPS PUB 186-4 Federal Information Processing Standards Publication Digital Signature Standard (DSS), July 2013
[FIPS PUB 198-1]	Federal Information Processing Standards Publication The Keyed-Hash Message Authentication Code (HMAC), July 2008
[800-90Arev1]	NIST Special Publication 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Revision 1, June 2015
[FIPS PUB 180-4]	FIPS PUB 180-4 Federal Information Processing Standards Publication Secure Hash Standard (SHS), August 2015