Assurance Activity Report for
CACI idt GoSilent Server and Cube v26.01

Collaborative Protection Profile for Network Devices Version 3.0e
***Collaborative Protection Profile for Network Devices*, Version 3.0e**
***PP-Module for VPN Gateways*, Version 1.3**

AAR Version 0.6, July 2026

Evaluated by:

2400 Research Blvd, Suite 395
Rockville, MD 20850

Prepared for:

National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:
CACI idt

The Author of the Security Target:
CACI idt

The TOE Evaluation was Sponsored by:
CACI idt

Evaluation Personnel:
Furukh Siddique
Alexander Fannin
Reema Nagwekar
Sagar Pujari

Common Criteria Version
Common Criteria Version 3.1 Revision 5

Common Evaluation Methodology Version
CEM Version 3.1 Revision 5

REVISION HISTORY

VERSION	DATE	CHANGES
0.1	July 30, 2025	Draft
0.2	October 20, 2025	Updated TSS and AGD activities
0.3	March 4, 2026	Minor Updates
0.4	May 20, 2026	Address Internal QA Feedback
0.5	June 24, 2025	Addressed ECR comments from Check-out
0.6	July 10, 2026	Minor Updates

CONTENTS

1 TOE OVERVIEW

The GoSilent Server acts as the centralized management and external access system for the TOE integrated into one system. The GoSilent Server provides the management GUI for configuration of GoSilent Server as well as the GoSilent Cube user devices. Operationally, it acts as the central peer for all IPsec connections from the GoSilent Cube devices and provides gateway connectivity to external systems located behind a GoSilent Cube.

The GoSilent Cube is a portable enterprise-grade firewall and VPN, ideal for sensitive communications, secure remote network access, and Internet of Things (IoT) deployments. GoSilent Cube can be setup within minutes by non-technical users. Physical Ethernet connections are supported on both the user side and VPN side of GoSilent Cube.

Together, GoSilent Cube and GoSilent Server provide a secure communications path to one or more systems located "behind" each GoSilent Cube. These systems connect to GoSilent Cube via physical Ethernet. Each GoSilent Cube establishes an IPsec VPN to the GoSilent Server, and all traffic from the user systems is routed over that VPN. Physical Ethernet is supported on the VPN side of GoSilent Cube.

The TOE applies policies to restrict the traffic that is permitted to pass between the user systems and external networks.

Management of the system is performed via a GUI provided by GoSilent Server, accessed via a browser on remote PCs using TLS/HTTPS. Authorized administrators may configure GoSilent Server as well as the GoSilent Cubes.

GoSilent Cubes also provide a GUI accessed from a browser on a user system via a TLS/HTTPS connection. This GUI only provides authorized administrators with the ability to configure that specific GoSilent Cube with enough information to connect to GoSilent Server. All additional configuration information is downloaded from GoSilent Server once the IPsec VPN is established.

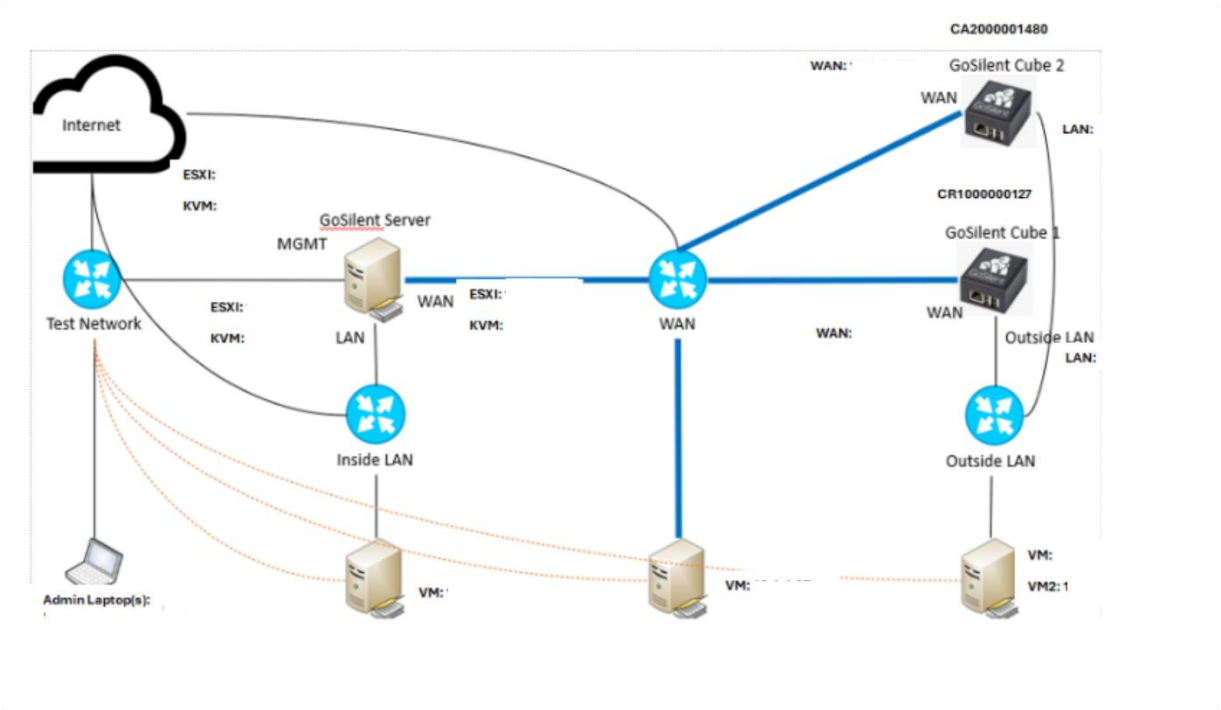
Both GoSilent Server and GoSilent Cube generate audit records that are stored locally as well as sent to a remote syslog server via a TLS connection.

2 ASSURANCE ACTIVITIES IDENTIFICATION

The Assurance Activities contained within this document include all those defined within the CPP_ND_V3.0E, MOD_CPP_FW_V1.4E and MOD_VPNGW_V1.3 based upon the core SFRs and those implemented based on selections within the PPs/EPs.

3 TEST BED DESCRIPTIONS

3.1 TEST BED



3.2 CONFIGURATION INFORMATION

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

Device Details Table					
Device Details		**Network Details**	**System Details**		
Device Name	**Function**	**Protocols**	**OS, including version**	**Timing Source**	**Software & Tools, including version**
TOE Name	TOE	SSH TLS 1.2, TLS 1.3	Centos 6	Manual	N/A
User Laptop	Tester's Laptop	SSH	Windows 10	NTP	Wireshark 11.2, OPENSSH (8.2p). OpenSSL (1.1.1f) Putty (Release 0.77) Hex editor (Version 2.5.0.0)
Ubuntu VM	Audit Server	TLS 1.2	Ubuntu 18.04	Manual	Wireshark 11.2, strongswan v1.0, acumen-tlsc-v2.2e, acumen-tlss-v2.2e, acumen-tlss, X509-mod
	Radius	TLS 1.3			

Pi Bridge	Bridge	N/A	Linux pi-gmc 5.15.61-v8+	Manual	N/A

3.3 TEST TIME AND LOCATION

All testing was carried out at Acumen Security offices located at 2400 Research Blvd Suite #395, Rockville, MD 20850. Testing occurred from May 2025 to July 2026.

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

4 DETAILED EVALUATION ACTIVITIES

4.1 TSS AND AGD ACTIVITIES

4.1.1 SECURITY AUDIT (FAU)

4.1.1.1 FAU_GEN.1 AUDIT DATA GENERATION (CPP_ND_V3.0E)

4.1.1.1.1 FAU_GEN.1 TSS

For the administrative task of generating/importing, 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.

Evaluator Findings:
The evaluator examined the TSS and ensured that it identifies what information is logged to identify the relevant cryptographic key during generating/import, changing, or deleting. The relevant information is found in the following section(s): TOE Summary Specification section 6.1.1 FAU_GEN.1, FAU_GEN.1/VPN, FAU_GEN.2, FAU_GEN_EXT.1" Upon investigation, the evaluator found that the TSS states that: **For audit records involving the generating/import of, changing, or deleting of cryptographic keys, the record identifies the key via reference to the certificate or key identifier associated with the key. For Post-quantum Pre-shared Keys (PPKs), the associated PPK-ID is included in the audit record.**

For distributed TOEs the evaluator shall examine the TSS and ensure that it describes which of the overall required auditable events defined in FAU_GEN.1.1 are generated and recorded by which TOE components.

Evaluator Findings:
The evaluator reviewed the TSS and ensured that it describes which of the overall required auditable events defined in FAU_GEN.1.1 are generated and recorded by which TOE components. The relevant information is found in the following section(s): TOE Summary Specification Specification section 6.1.1 "FAU_GEN.1, FAU_GEN.1/VPN, FAU_GEN.2, FAU_GEN_EXT.1, FAU_STG_EXT.1 & FAU_STG_EXT.4" Upon investigation, the evaluator found that the TSS states that: **The GoSilent Server and GoSilent Cube components generate audit records associated with the SFRs they implement as specified in Table 18 of the ST. This table has been iterated below:**

SFR	Dist. Requirement	GoSilent Server	GoSilent Cube
FAU_GEN.1	All	X	X
FAU_GEN.1/VPN	All	X	X
FAU_GEN.2	All	X	X
FAU_GEN_EXT.1	All	X	X
FAU_STG_EXT.1	All	X	X
FAU_STG_EXT.4	Feature Dependent	X	X
FCO_CPC_EXT.1	All	X	X
FCS_CKM.1	One	X	X

FCS_CKM.1/IKE	One	X	X
FCS_CKM.2	All	X	X
FCS_CKM.4	All	X	X
FCS_COP.1/DataEncryption	All	X	X
FCS_COP.1/SigGen	All	X	X
FCS_COP.1/Hash	All	X	X
FCS_COP.1/KeyedHash	All	X	X
FCS_HTTPS_EXT.1	Feature Dependent	X	X
FCS_IPSEC_EXT.1	Feature Dependent	X	X
FCS_RBG_EXT.1	All	X	X
FCS_TLSC_EXT.1	Feature Dependent	X	X
FCS_TLSS_EXT.1	Feature Dependent	X	X
FDP_RIP.2	Feature Dependent	X	X
FFW_RUL_EXT.1	One	X	
FIA_AFL.1	One	X	X
FIA_PMG_EXT.1	One	X	X
FIA_PSK_EXT.1	Feature Dependent	X	X
FIA_PSK_EXT.2	Feature Dependent	X	X
FIA_UIA_EXT.1	One	X	X
FIA_UAU.7	Feature Dependent	X	X
FIA_X509_EXT.1/ITT	Feature Dependent	X	X
FIA_X509_EXT.1/Rev	Feature Dependent	X	X
FIA_X509_EXT.2	Feature Dependent	X	X
FIA_X509_EXT.3	Feature Dependent	X	X
FMT_MOF.1/ManualUpdate	All	X	X
FMT_MOF.1/Services	Feature Dependent	X	X
FMT_MTD.1/CoreData	All	X	X
FMT_MTD.1/CryptoKeys	Feature Dependent	X	X
FMT_SMF.1	Feature Dependent	X	X
FMT_SMF.1/FFW	Feature Dependent	X	
FMT_SMF.1/VPN	Feature Dependent	X	
FMT_SMR.2	One	X	X
FPF_RUL_EXT.1	Feature Dependent	X	
FPT_APW_EXT.1	Feature Dependent	X	X
FPT_FLS.1/SelfTest	All	X	X
FPT_ITT.1	Feature Dependent	X	X
FPT_SKP_EXT.1	All	X	X
FPT_STM_EXT.1	All	X	X
FPT_TST_EXT.1	All	X	X
FPT_TST_EXT.3	All	X	X
FPT_TUD_EXT.1	All	X	X

FTA_SSL_EXT.1	Feature Dependent	X	X
FTA_SSL.3	Feature Dependent	X	X
FTA_SSL.4	Feature Dependent	X	X
FTA_TAB.1	One	X	X
FTP_ITC.1	One	X	X
FTP_ITC.1/VPN	One	X	X
FTP_TRP.1/Admin	One	X	X

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

Evaluator Findings:

The evaluator examined the TSS and ensured that the mapping of audit events to TOE components accounts for, and is consistent with, information provided in Table 1, as well as events in Tables 2, 4, and 5 (as applicable to the overall TOE). The evaluator confirmed that all components defined as generating audit information for a particular SFR contributed to that SFR as defined in the mapping of SFRs to TOE components, and that the audit records generated by each component covered all the SFRs that it implements.

The relevant information is found in the following section(s): TOE Summary Specification section 6.1.1 labeled "FAU_GEN.1, FAU_GEN.1/VPN, FAU_GEN.2, FAU_GEN_EXT.1, FAU_STG_EXT.1 & FAU_STG_EXT.4"

Upon investigation, the evaluator found that the TSS states that: **The specific events audited on each component correspond to the SFRs applicable to each component and can be found in Table 11 and Table 12 in the ST.**

The GoSilent Server and GoSilent Cube components generate audit records associated with the SFRs they implement as specified in Table 10 of the ST.

The information in all three charts have been verified to map accurately to the SFRs and the components claimed/used by the TOE.

Verdict:
PASS.

4.1.1.1.2 FAU_GEN.1 AGD

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).

Evaluator Findings:
The evaluator checked the guidance documentation and ensured 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, was provided from the actual audit record). The relevant information is found in the following section(s): **"Log Reference"** Upon investigation, the evaluator found that the AGD activity states that: **Table 3: Audit Events in the AGD provides an example audit for each auditable event.**

The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes.

Evaluator Findings:
The evaluator made a determination of the administrative actions related to TSF data related to configuration changes. The relevant information is found in the following section(s): **"Format"** under **"Log Reference"** Upon investigation, the evaluator found that the AGD activity states that**: Each audit record includes the following fields:** **a. Date/Timestamp** **b. Event Type** **c. Subject Identity** **d. Event Outcome (Success/Failure)**

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.

Evaluator Findings:
The evaluator examined the guidance documentation and made 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 relevant information is found in the following section(s): **"Audit Logging"** **The evaluator performed this activity as a part of those Assurance Activities associated with ensuring the corresponding guidance documentation satisfies their independent requirements. Overall, the evaluator considered the administrator guides published by the vendor. The evaluator reviewed the contents of these documents and looked specifically for functionality related to the scope of the evaluation.**

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.

Evaluator Findings:
The evaluator documented 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. The relevant information is found in the following section(s): **"Audit Logging"** **The evaluator performed this activity as a part of those Assurance Activities associated with ensuring the corresponding guidance documentation satisfies their independent requirements. Overall, the evaluator considered the administrator guides published by the vendor. The evaluator reviewed the contents of these documents and looked specifically for functionality related to the scope of the evaluation.**

Verdict:
PASS.

4.1.1.2 FAU_GEN_EXT.1 SECURITY AUDIT DATA GENERATION FOR DISTRIBUTED TOE COMPONENTS (CPP_ND_V3.0E)

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

Evaluator Findings:
The evaluator examined the TSS and AGD and ensured that the requirements on TSS, AGD and Tests regarding FAU_GEN_EXT.1 are already covered by the corresponding requirements for FAU_GEN.1. The relevant information is found in the following section(s): **TOE Summary Specification section 6.1.1 labeled "FAU_GEN.1, FAU_GEN.1/VPN, FAU_GEN.2, FAU_GEN_EXT.1"**. Upon investigation, the evaluator found that the TSS states that**: The TOE generates the audit records specified at FAU_GEN.1 containing fields that include the timestamp, IP address (if applicable), action, user (if applicable) and a contextual message indicating success or failure of the action.** **GoSilent Server and GoSilent Cube generate audit records specified at FAU_GEN.1, FAU_GEN.1/VPN, FAU_GEN.2, and FAU_GEN_EXT.1 for operation and administration of each component. The specific events audited on each component correspond to the SFRs applicable to each component and can be found in Table 11 and Table 12. Auditing is always enabled, and each audit record includes:** • **Date and time of the event,** • **Type (i.e., category and action) of event,** • **Subject (i.e., user and domain) identity,** • **Result (success or failure) of the event, and** • **Description (where applicable access mode, target object, etc.).** **The TOE includes the user identity in audit events resulting from actions of identified users.** **For audit records involving the generating/import of, changing, or deleting of cryptographic keys, the record identifies the key via reference to the certificate or key identifier associated with the key. For Post-quantum Pre-shared Keys (PPKs), the associated PPK-ID is included in the audit record.** **The GoSilent Server and GoSilent Cube components generate audit records associated with the SFRs they implement as specified in Table 17.**

This information covers the FAU_GEN_EXT.1 requirements for the TSS in full.

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

Upon investigation, the evaluator found that the AGD activity states that: Each audit record includes the following fields:

a) Date/Timestamp

b) Event Type

c) Subject Identity

d) Event Outcome (Success/Failure)

Verdict:

PASS.

4.1.1.3 FAU_STG_EXT.1 PROTECTED AUDIT EVENT STORAGE (CPP_ND_V3.0E)

4.1.1.3.1 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.

Evaluator Findings:
The evaluator examined the TSS and ensured that it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided. The relevant information is found in the following section(s): **TOE Summary Specification section 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4"** Upon investigation, the evaluator found that the TSS states that: **The audit data transferred to the external syslog server consists of each audit record generated on the GoSilent components. Each component transmits a copy of each audit record to an external syslog server in real time. The syslog server is configured on GoSilent Server and communicated to all GoSilent Cubes.** **The trusted channel for transferring the audit records is provided by TLS (Transport Layer Security) through the established IPsec tunnel. Once the IPsec header has been stripped by GoSilent Server, the syslog (TLS) traffic is forwarded to the syslog server based on the IP destination address.**

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.

Evaluator Findings:
The evaluator examined the TSS Specification section 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4" and ensured that 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 relevant information is found in the following section(s): **TOE Summary Specification section 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4."**

Upon investigation, the evaluator found that the TSS states that: **The TOE is a distributed TOE that stores audit data locally on each TOE component. Each TOE component stores audit event records that are generated on that component.**

Each TOE component persistently stores audit event records that are generated on that component. GoSilent Server can store up to 50MB of audit information and the Cube can store up to 1.5MB. When that space is exhausted, the oldest records are discarded so that new records can be saved. The records are stored in a series of 5 files, 10MB each on GoSilent Server and 300KB each on the Cube. The current file is always "messages". When it fills, it is initially renamed messages1 and a new (empty) messages file is created. As additional files fill, the number appended to each of the existing files is incremented to make room for a new messages1 file. The name of the messages4 file is not incremented; instead, that file is deleted to limit the amount of saved data.

The evaluator shall examine the TSS to ensure that for distributed TOEs it contains a list of TOE components that store audit data locally.

Evaluator Findings:
The evaluator examined the TSS and ensured that for distributed TOEs it contains a list of TOE components that store audit data locally. The relevant information is found in the following section(s): **TOE Summary Specification section 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4** Upon investigation, the evaluator found that the TSS states that: **The TOE is a distributed TOE that persistently stores audit data locally on each TOE component. GoSilent Server can store up to 50MB of audit information and the Cube can store up to 1.5MB**

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

Evaluator Findings:
The evaluator examined the TSS and ensured that for distributed TOEs that contain components which do not store audit data locally but transmit their generated audit data to other components it contains a mapping between the transmitting and storing TOE components. The relevant information is found in the following section(s): **TOE Summary Specification section 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4** Upon investigation, the evaluator found that the TSS states that: **There are no TOE components that do not contain audit data. This activity is considered satisfied. This is determined by the information in the TSS that states, "Each TOE component persistently stores audit event records that are generated on that component."**

The evaluator shall examine the TSS to ensure that it details whether the transmission of audit data to an external IT entity can be done in real-time, periodically, or both. In the case where the TOE is capable of performing transmission periodically, the evaluator shall 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.

Evaluator Findings:
The evaluator examined the TSS and ensured that it details whether the transmission of audit information to an external IT entity can be done in real-time, periodically, or both. In the case where the TOE is capable of performing transmission periodically, the evaluator verified 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. The relevant information is found in the following section(s): **TOE Summary Specification 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4** Upon investigation, the evaluator found that the TSS states that: **Each component also transmits a copy of each audit record to an external syslog server in real time, the syslog server is configured on GoSilent Server and communicated to all GoSilent Cubes.**

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

Evaluator Findings:
The evaluator examined the TSS and ensured that it describes to which TOE components this SFR applies and how audit data transfer to the external audit server is implemented among the different TOE components (e.g. every TOE components does its own transfer or the data is sent to another TOE component for central transfer of all audit events to the external audit server). The relevant information is found in the following section(s): **TOE Summary Specification 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4."** Upon investigation, the evaluator found that the TSS states that: **Each component also transmits a copy of each audit record to an external syslog server in real time; the syslog server is configured on GoSilent Server and communicated to all GoSilent Cubes. Each Cube transmits the syslog server traffic via TLS through the established IPsec tunnel. Once the IPsec header has been stripped by GoSilent Server, the syslog (TLS) traffic is then forwarded to the syslog server based on the IP destination address. If the connection to the syslog server is unavailable, audit records continue to be saved locally. Once the connection is re-established, audit logs saved locally but not previously transmitted to the syslog server and sent.**

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

Evaluator Findings:
The evaluator examined the TSS and ensured it describes the amount of audit data that are stored locally and how these records are protected against unauthorized modification or deletion. The relevant information is found in the following section(s): **TOE Summary Specification section 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4.".** Upon investigation, the evaluator found that the TSS states that: **The TOE is a distributed TOE that persistently stores audit data locally on each TOE component. GoSilent Server can store up to 50MB of audit information and the Cube can store up to 1.5MB. When that space is exhausted, the oldest records are discarded so that new records can be saved. The records are stored in a series of 5 files, 10MB each on GoSilent Server and 300KB each on the Cube. The current file is always "messages". When it fills, it is initially renamed messages1 and a new (empty) messages file is created. As additional files fill, the number appended to each of the existing files is incremented to make room for a new messages1 file. The name of the messages4 file is not incremented; instead, that file is deleted to limit the amount of saved data.** **Only authorized administrators may view audit records and no capability to modify the audit records is provided.**

The evaluator shall examine the TSS to ensure it describes the method implemented for local logging, including format (e.g. buffer, log file, database) and whether the logs are persistent or non-persistent.

Evaluator Findings:
The evaluator examined the TSS and ensured it describes the method implemented for local logging, including format (e.g. buffer, log file, database) and whether the logs are persistent or non-persistent. The relevant information is found in the following section(s): **TOE Summary Specification section 6.1.1 "FAU_GEN.1, FAU_GEN.1/VPN, FAU_GEN.2, FAU_GEN_EXT.1" and 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4."** Upon investigation, the evaluator found that the TSS states that: **6.1.1 states that: GoSilent Server and GoSilent Cube generate audit records specified in FAU_GEN.1, FAU_GEN.1/VPN, FAU_GEN.2, and FAU_GEN_EXT.1 for operation and administration of each component. The specific events audited on each component correspond to the SFRs applicable to each component and can be found in** Error! Reference source not found. **and** Error! Reference source not found.**. Auditing is always enabled, and each audit record includes:** • **Date and time of the event,** • **Type (i.e., category and action) of event,** • **Subject (i.e., user and domain) identity,** • **Result (success or failure) of the event, and** • **Description (where applicable access mode, target object, etc.).** **The TOE includes the user identity in audit events resulting from actions of identified users.** **6.1.2 states that: Each component also transmits a copy of each audit record to an external syslog server in real time; the syslog server is configured on GoSilent Server and communicated to all GoSilent Cubes. Each Cube transmits the syslog server traffic via TLS through the established IPsec tunnel. Once the IPsec header has been stripped by GoSilent Server, the syslog (TLS) traffic is then forwarded to the syslog server based on the IP destination address. If the connection to the syslog server is unavailable, audit records continue to be saved locally. Once the connection is re-established, audit logs saved locally but not previously transmitted to the syslog server and sent.**

The evaluator shall examine the TSS to ensure it describes the conditions that must be met for authorized deletion of audit records.

Evaluator Findings:
The evaluator examined the TSS and ensured it describes the conditions that must be met for authorized deletion of audit records. The relevant information is found in the following section(s): **TOE Summary Specification 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4.".** Upon investigation, the evaluator found that the TSS states that**: Only authorized administrators may view audit records and no capability to modify the audit records is provided.** **When a preconfigured space (50MB for the GoSilent Server and 1.5MB for the Cube) is exhausted the oldest records are deleted.**

The evaluator shall examine the TSS to ensure 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.

Evaluator Findings:
The evaluator examined the TSS and ensured 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 is detailed in the TSS. The relevant information is found in the following section(s): **TOE Summary Specification 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4.".** Upon investigation, the evaluator found that the TSS states that: **The TOE is a distributed TOE that persistently stores audit data locally on each TOE component. GoSilent Server can store up to 50MB of audit information and the Cube can store up to 1.5MB. When that space is exhausted, the oldest records are discarded so that new records can be saved. The records are stored in a series of 5 files, 10MB each on GoSilent Server and 300KB each on the Cube. The current file is always "messages". When it fills, it is initially renamed messages1 and a new (empty) messages file is created. As additional files fill, the number appended to each of the existing files is incremented to make room for a new messages1 file. The name of the messages4 file is not incremented; instead, that file is deleted to limit the amount of saved data.**

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

Evaluator Findings:
The evaluator examined the TSS and ensured that it describes which TOE components are storing audit information locally and which components are buffering audit information and forwarding the information to another TOE component for local storage. For every component the TSS describes the behaviour when local storage space or buffer space is exhausted. The relevant information is found in the following section(s): **TOE Summary Specification 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4."** Upon investigation, the evaluator found that the TSS states that: **The TOE is a distributed TOE that persistently stores audit data locally on each TOE component. GoSilent Server can store up to 50MB of audit information and the Cube can store up to 1.5MB. When that space is exhausted, the oldest records are discarded so that**

new records can be saved. The records are stored in a series of 5 files, 10MB each on GoSilent Server and 300KB each on the Cube. The current file is always "messages". When it fills, it is initially renamed messages1 and a new (empty) messages file is created. As additional files fill, the number appended to each of the existing files is incremented to make room for a new messages1 file. The name of the messages4 file is not incremented; instead, that file is deleted to limit the amount of saved data.

Verdict:

PASS.

4.1.1.3.2 FAU_STG_EXT.1 AGD

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.

Evaluator Findings:
The evaluator examined the guidance documentation **"Cryptographic Functions"** and ensured 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. Upon investigation, the evaluator found that the AGD states that: GoSilent Server: **To modify the logging parameters of the TOE, log into the GSS GUI and navigate to the following:** **'System Settings > Logging'** **and scroll down to each of the following settings:** **o 'Enable log shipping' – Must be enabled.** **o 'Log destination host' – Must be configured.** **o 'Log shipping port' – Must be configured.** **o 'Log shipping protocol' – Must be set to TCP.** **o 'Log shipping TLS cipher' – Default is TLS 1.2: AES128-GCM-SHA256** **Must be configured to one of:** **TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384** **TLS 1.3: TLS_AES_256_GCM_SHA384** GoSilent Cube: **Each Cube will transmit audit log traffic via TLS through the established IPsec tunnel. The IPSec header is then stripped by GSS and traffic is processed by the firewall ruleset which forwards syslog traffic to the syslog server. GSS initiates the connection to the syslog server at startup, and the Cube will initiate a connection to the syslog server once the IPSec tunnel has been successfully established.** **The TOE only supports TLSv1.2 and TLSv1.3 when acting as a TLS client to provide a trusted channel to the Syslog server. The Cube will always initiate the connection using one of the following ciphersuites:** **TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384** **TLS 1.3: TLS_AES_256_GCM_SHA384** **This ciphersuite can be configured as described in sections 3.11.** **Note: This change will take effect after the user has disconnected and reconnected to the GSS server profile on the Cube Console. The GSS server profile must be set as the default server profile.**

The evaluator shall also examine the guidance documentation to ensure 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.

Evaluator Findings:
The evaluator also examined the guidance documentation **"Audit Logging"** and determined that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. Upon investigation, the evaluator found that the AGD states that: **Each TOE component also transmits a copy of each audit record to an external syslog server in real time via TLS.** **Each Cube will transmit audit log traffic via TLS through the established IPsec tunnel. The IPSec header is then stripped by GSS and traffic is processed by the firewall ruleset which forwards syslog traffic to the syslog server. GSS initiates the connection to the syslog server at startup, and the Cube will initiate a connection to the syslog server once the IPSec tunnel has been successfully established.**

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

Evaluator Findings:
The evaluator examined the guidance documentation **"Audit Logging"** and ensured it describes any configuration required for protection of the locally stored audit data against unauthorized modification or deletion. Upon investigation, the evaluator found that the AGD states that: **Only authorized administrators may view these audit records (locally or remotely), and no capability to modify or delete the audit record is provided.**

If the storage size is configurable, the evaluator shall review the Guidance Documentation to ensure it contains instructions on specifying the required parameters.

Evaluator Findings:
If the storage size is configurable, the evaluator reviewed the Guidance Documentation "**Audit Logging**" to ensure it contains instructions on specifying the required parameters. **Each TOE component stores its own audit records locally with up to 50MB of audit information across five 10MB files on GSS and up to 1.5MB of audit information across five 300KB files on the Cube.**

If more than one selection is made for FAU_STG_EXT.1.5, the evaluator shall review the Guidance Documentation to ensure it contains instructions on specifying which action is performed when the local storage space is full.

Evaluator Findings:
If more than one selection is made for FAU_STG_EXT.1.5, the evaluator reviewed the Guidance Documentation **"Audit Logging"** to ensure it contains instructions on specifying which action is performed when the local storage space is full. **Each TOE component stores its own audit records locally with up to 50MB of audit information across five 10MB files on GSS and up to 1.5MB of audit information across five 300KB files on the Cube. Both components will delete the oldest records first during log rotation.**

Verdict:

PASS.

4.1.1.4 FAU_STG_EXT.4 PROTECTED LOCAL AUDIT EVENT STORAGE FOR DISTRIBUTED TOES & FAU_STG_EXT.5 PROTECTED REMOTE AUDIT EVENT STORAGE FOR DISTRIBUTED TOES (CPP_ND_V3.0E)

4.1.1.4.1 FAU_STG_EXT.4 TSS

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

Evaluator Findings:
The evaluator examined the TSS and confirmed it describes which TOE components store their security audit events locally and which send their security audit events to other TOE components for local storage. For the latter, the target TOE component(s) which store security audit events for other TOE components are identified. For every sending TOE component, the corresponding receiving TOE component(s) need to be identified. For every transfer of audit information between TOE components a description was provided as to the data is secured during transfer according to FTP_ITC.1 or FPT_ITT.1. The relevant information is found in the following section(s): **TOE Summary Specification 6.1.2 labeled "FAU_STG_EXT.1 & FAU_STG_EXT.4."** Upon investigation, the evaluator found that the TSS states that: **The TOE is a distributed TOE that persistently stores audit data locally on each TOE component. The records are stored in a series of 5 files, 10MB each on GoSilent Server and 300KB each on the Cube.** **Each component also transmits a copy of each audit record to an external syslog server in real time; the syslog server is configured on GoSilent Server and communicated to all GoSilent Cubes. Each Cube transmits the syslog server traffic via TLS through the established IPsec tunnel. Once the IPsec header has been stripped by GoSilent Server, the syslog (TLS) traffic is then forwarded to the syslog server based on the IP destination address. If the connection to the syslog server is unavailable, audit records continue to be saved locally. Once the connection is re-established, audit logs saved locally but not previously transmitted to the syslog server and sent.** **GoSilent Server provides a mechanism for authorized administrators to view local audit records via the management GUI. GoSilent provides the capability to export audit record files for external review.**

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

Evaluator Findings:
N/A. All TOE components store data locally.

Verdict:

PASS.

4.1.1.4.2 FAU_STG_EXT.4 AGD

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

Evaluator Findings:
The evaluator examined the guidance documentation and ensured that it describes how the link between different TOE components is established if audit data is exchanged between TOE components for local storage. The guidance documentation also describes all possible configuration options for local storage of audit data and provides all instructions how to perform the related configuration of the TOE components. The relevant information is found in the following section(s): **Audit Logging** Upon investigation, the evaluator found that the AGD activity states that: **Each TOE component stores its own audit records locally with up to 50MB of audit information across five 10MB files on GSS and up to 1.5MB of audit information across five 300KB files on the Cube. Both components will delete the oldest records first during log rotation.** **Each TOE component also transmits a copy of each audit record to an external syslog server in real time via TLS. Syslog server parameters are configured on the GoSilent Server and communicated to all GoSilent Cubes.** **Each Cube will transmit audit log traffic via TLS through the established IPsec tunnel. The IPSec header is then stripped by GSS and traffic is processed by the firewall ruleset which forwards syslog traffic to the syslog server. GSS initiates the connection to the syslog server at startup, and the Cube will initiate a connection to the syslog server once the IPSec tunnel has been successfully established.**

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

Evaluator Findings:
N/A. All TOE components store data locally.

Verdict:

PASS.

4.1.2 COMMUNICATION (FCO)

4.1.2.1 FCO_CPC_EXT.1 COMPONENT REGISTRATION CHANNEL DEFINITION (CPP_ND_V3.0E)

4.1.2.1.1 FCO_CPC_EXT.1 TSS

The evaluator shall examine the TSS to confirm that it:

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

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

- First type: the TSS identifies the relevant SFR iteration that specifies the channel used
- Second type: the TSS (with support from the AGD if selected in FTP_TRP.1.3/Join) describes details of the channel and the mechanisms that it uses (and describes how the process ensures that the key is unique to the pair of components) – see also the Evaluation Activities for FTP_TRP.1/Join.

Evaluator Findings:
The evaluator examined the TSS and confirmed that it: a) Describes the method by which a Security Administrator enables and disables communications between pairs of TOE components. b) Describes the relevant details according to the type of channel in the main selection made in FCO_CPC_EXT.1.2: • First type: the TSS identifies the relevant SFR iteration that specifies the channel used • Second type: the TSS (with support from the AGD if selected in FTP_TRP.1.3/Join) describes details of the channel and the mechanisms that it uses (and describes how the process ensures that the key is unique to the pair of components) – see also the Evaluation Activities for FTP_TRP.1/Join. The relevant information is found in the following section(s): **TOE Summary Specification section 6.2.1 "FCO_CPC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **GoSilent Cube registration (enablement) involves some manual configuration by an authorized administrator on GoSilent Server prior to the enablement process taking place. The X.509 certificate for GoSilent Server and the specific GoSilent Cube are communicated out of band and imported by an authorized administrator on that component. Guidance information instructs administrators to use a secure communication channel for communication of the certificates.** **When the necessary steps have been taken by an administrator to prepare the Cube for enablement, the channel described in FCO_CPC_EXT.1 and FTP_ITC.1/VPN is used to execute and finalize the registration process which occurs during the first connection of a Cube to GoSilent Server.** **Any attempt by a GoSilent Cube to establish an IPsec tunnel (connect) to GoSilent Server is rejected by the GoSilent Server if enablement has not been completed. Each GoSilent Cube only establishes an IPsec tunnel with GoSilent Server. Attempts to establish an IPsec tunnel from one GoSilent Cube to another are always rejected. Alternatively, a cube may also be suspended from within the GoSilent Server GUI, whereby subsequent connections from the cube are rejected.**

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

Evaluator Findings:
This activity is not applicable as there are no aspects of the registration channel that do not meet FTP_ITC or FPT_ITT.

Verdict:

PASS.

4.1.2.1.2 FCO_CPC_EXT.1 AGD

The evaluator shall examine the AGD to confirm that it contains instructions for enabling and disabling communications with any individual component of a distributed TOE.

Evaluator Findings:
The evaluator examined the AGD and confirmed that it contains instructions for enabling and disabling communications with any individual component of a distributed TOE. The relevant information is found in the following section(s): "**Installation and Verification**" Upon investigation, the evaluator found that the AGD activity states that: **Each GoSilent Cube must be registered on the GoSilent Server. Registration involves several steps to be completed by an administrator prior to a Cube establishing a connection to GSS. These steps can be completed by navigating to the 'VPN Clients' tab in GSS and clicking on the 'Activate' button next to a listed cube that has been licensed for GSS. The 'Add A Client' window will appear where the 'Device Serial Number' is listed but not configurable. Fill in the 'Client ID' and 'Client Description' fields and click 'Add Client'. Once these steps to prepare the Cube for registration have been completed, a connection attempt from the Cube to GSS can be made.** **Note: The first connection attempt from the Cube to GSS will complete the enablement process and may take longer to establish than subsequent connection attempts.**

The evaluator shall confirm that the method of disabling is such that all other components can be prevented from communicating with the component that is being removed from the TOE (preventing the remaining components from either attempting to initiate communications to the disabled component, or from responding to communications from the disabled component).

Evaluator Findings:
The evaluator confirmed that the method of disabling is such that all other components can be prevented from communicating with the component that is being removed from the TOE (preventing the remaining components from either attempting to initiate communications to the disabled component, or from responding to communications from the disabled component). The relevant information is found in the following section(s): "**Installation and Verification**" Upon investigation, the evaluator found that the AGD activity states that: **Cubes may also be suspended and thereby subsequent connections from a cube can be rejected, by also navigating to the *'VPN Clients'* tab in GSS and clicking on the pencil icon beside the Cube to be suspended, toggle the "*Suspend*" button, and click "*Save*".**

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

Evaluator Findings:
The evaluator examined the AGD and confirmed that it includes recovery instructions should a connection be unintentionally broken during the registration process. The relevant information is found in the following section(s): **"Installation and Verification"** Upon investigation, the evaluator found that the AGD activity states that: **Should the enablement process not complete successfully, the administrator can remove the Cube in GSS by navigating to '*VPN Clients'* and clicking the trash can icon ('*Remove'*) next to the Cube experiencing issues, reboot the Cube device, and restarting the Enablement process again.**

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

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

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

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

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

Evaluator Findings:
If the TOE uses a registration channel for registering components to the TOE (i.e. where the ST author uses the FTP_ITC.1/FPT_ITT.1 or FTP_TRP.1/Join channel types in the main selection for FCO_CPC_EXT.1.2) then the evaluator examined the Preparative Procedures and confirmed that they: a) describe the security characteristics of the registration channel (e.g. the protocol, keys and authentication data on which it is based) and highlights any aspects which do not meet the requirements for a steady- state inter-component channel (as in FTP_ITC.1 or FPT_ITT.1) b) identify any dependencies between the configuration of the registration channel and the security of the subsequent inter-component communications (e.g. where AES-256 inter-component communications depend on transmitting 256 bit keys between components and therefore rely on the registration channel being configured to use an equivalent key length) c) identify any aspects of the channel can be modified by the operational environment in order to improve the channel security and describes how this modification can be achieved (e.g. generating a new key pair, or replacing a default public key certificate).

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

Upon investigation, the evaluator found that the AGD activity states that: **GoSilent Cubes always initiate the connection to establish an IPsec tunnel with GSS.**

To modify the VPN parameters of the TOE, log into the GSS GUI as described in section 3.1 and navigate to the following:

'*System Settings > VPN*'

and scroll down to each of the following settings:

- **'*IPSec Protocol Mode*' – Set to "IKEv2_Certificates"**
- **'*IPSec Cipher Suite*' – Default is aes-256gcm16-pfsha512-ecp521 (Top Secret DH21)**

  Can configure to:

  aes-256gcm16-pfsha384-ecp384 (Top Secret DH20)
- **'*Security Association (SA) Lifetime*' – Between 1 and 24 hours**
- **'*Child SA Lifetime*' – Between 1 and 8 hours**
- **'*Enable Strict OCSP Checking*' – Must be enabled**

The GoSilent Cube always initiates the connection to establish a tunnel with GSS. Before the Cube can initiate this connection, a Virtual Server profile containing the GSS server information must be configured on the Cube. This can be done by logging into the Cube by navigating to the following:

Settings > Server Profiles

Click the '*Add*' button next to 'Virtual Server' and enter the details for GSS.

Click '*Add Virtual Server'.* The newly configured server profile for GSS will be shown in the Server Profiles list.

Note: Ensure that all GSS VPN configurations and Cube enablement have been completed as described above.

Click the '*Connect'* button on the GSS server profile. In the '*Are you sure?*' window click '*Yes, I'm Sure'.*

The GSS server profile will show the status '*Connected*'

The '*GoSilent Status'* shown in the top left of the Cube GUI will show '*VPN Connected*'.

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

Evaluator Findings:
This evaluation does not make a claim for FTP_TRP.1/Join. This activity is not applicable.

Verdict:

PASS.

4.1.3 CRYPTOGRAPHIC SUPPORT (FCS)

4.1.3.1 FCS_CKM.1 CRYPTOGRAPHIC KEY GENERATION (CPP_ND_V3.0E)

4.1.3.1.1 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.

Evaluator Findings:
The evaluator ensured that the TSS identifies the key sizes supported by the TOE. The evaluator examined the TSS to verify that it identifies the usage for each scheme. The relevant information is found in the following section(s): **TOE Summary Specification section 6.1.3 FCS_CKM.1, FCS_CKM.1/IKE, & FCS_CKM.2** Upon investigation, the evaluator found that the TSS states that**: As described in FCS_CKM.1, the TOE generates P-384 and P-521 ECC keys in support of TLS client connections with the syslog server, and P-384 and P-521 ECC keys in support of Administrative GUI TLS server sessions. When performing Elliptic Curve Diffie-Hellman, the TOE acts as a sender with the TLS client connections and a recipient with the TLS server connections.** **These keys are also used to authenticate the TOE component to the administrator in TLS exchanges for the management GUI.** **As described in FCS_CKM.1/IKE, the TOE generates ECDSA P-384 and P-521 Elliptic Curve keys as specified in FIPS Pub 186-4 "Digital Signature Standard (DSS)" Appendix B.4 and implements all "shall" and "should" statements and does not implement any "shall not" or "should not" statements."**

Verdict:

PASS.

4.1.3.1.2 FCS_CKM.1 AGD

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

Evaluator Findings:
The evaluator verified that the AGD instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all cryptographic protocols defined in the Security Target. The relevant information is found in the following section(s): **"IPsec VPN Configuration"**, "**Cryptographic Function**" & **"Audit Logging"** Upon investigation, the evaluator found that the AGD activity states that: **To modify the VPN parameters of the TOE, log into the GSS GUI as described in section 3.1 and navigate to the following:** **'System Settings > VPN'** **and scroll down to each of the following settings:** **o 'IPSec Protocol Mode' – Set to "IKEv2_Certificates"** **o 'IPSec Cipher Suite' – Default is aes-256gcm16-pfsha512-ecp521 (Top Secret DH21)** **Can configure to:** **aes-256gcm16-pfsha384-ecp384 (Top Secret DH20)** **The TOE performs cryptographic functionality in support of authentication and logging actions. The following key establishment parameters are implemented by the TOE:** a) **TLS Client to syslog server – Elliptic-curve (P-384)** b) **TLS Server for administrative GUI - Elliptic-curve (P-384, P-521)** c) **IKE/IPsec - Elliptic-curve (P-384, P-521)** d) **PPKs** **The TOE performs cryptographic signature generation and verification services using ECDSA P-384 and P-521 in support of IPsec, TLS, X509, and trusted update functions. Additional information on certificate parameters can be found in section called "Certificate Management."** **To modify the logging parameters of the TOE, log into the GSS GUI and navigate to the following:** **'System Settings > Logging'** **and scroll down to each of the following settings:** **o 'Enable log shipping' – Must be enabled.** **o 'Log destination host' – Must be configured.** **o 'Log shipping port' – Must be configured.** **o 'Log shipping protocol' – Must be set to TCP.** **o 'Log shipping TLS cipher' – Default is TLS 1.2: AES128-GCM-SHA256** **Must be configured to one of:** **TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384** **TLS 1.3: TLS_AES_256_GCM_SHA384**

Verdict:

PASS.

4.1.3.2 FCS_CKM.2 CRYPTOGRAPHIC KEY ESTABLISHMENT (CPP_ND_V3.0E)

4.1.3.2.1 FCS_CKM.2 TSS

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 as shown in the table below. The information provided in this example 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.

Evaluator Findings:
The evaluator ensured that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. The evaluator examined 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 relevant information is found in the following section(s): **TOE Summary Specification section 6.1.3 FCS_CKM.1, FCS_CKM.1/IKE, & FCS_CKM.2** Upon investigation, the evaluator found that the TSS states that: **The TOE acts as a sender and a recipient when performing Diffie-Hellman.** **The TOE supports key establishment as follows:** **TLS Client to syslog server – Elliptic-curve (P-384 and P-521)** **TLS Server for administrative GUI - Elliptic-curve (P-384 and P-521)** **IKE/IPsec - Elliptic-curve (P-384 and P-521)** **ECDSA key pairs for X509 certificates can also be generated through the TOE GUI where the option of keys using P-384 and P-521 curves can be selected.**

Verdict:

PASS.

4.1.3.2.2 FCS_CKM.2 AGD

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

Evaluator Findings:

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

The relevant information is found in the following section(s): **"VPN", "Cryptographic Functions" & "Audit Logging"**

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

To modify the VPN parameters of the TOE, log into the GSS GUI as described in section 3.1 and navigate to the following:

'System Settings > VPN'

and scroll down to each of the following settings:

- **'IPSec Protocol Mode' – Set to "IKEv2_Certificates"**
- **'IPSec Cipher Suite' – Default is aes-256gcm16-pfsha512-ecp521 (Top Secret DH21)**

Can configure to:

aes-256gcm16-pfsha384-ecp384 (Top Secret DH20)

The TOE performs cryptographic functionality in support of authentication and logging actions. The following key establishment parameters are implemented by the TOE:

a) **TLS Client to syslog server – Elliptic-curve (P-384)**

b) **TLS Server for administrative GUI - Elliptic-curve (P-384, P-521)**

c) **IKE/IPsec - Elliptic-curve (P-384, P-521)**

The TOE performs cryptographic key generation services during IPsec tunnel establishment for IKE peer authentication using ECDSA P-384 and P-521 Elliptic curve keys as specified in FIPS Pub 186-5 "Digital Signature Standard (DSS)" Appendix B.4. These keys are used in the Diffie-Hellman process for IPsec and produce key sizes equivalent to or greater than 192 bits. Additional information on the specification of algorithms used for IPsec connections can be found in Section 3.8.

The TOE only supports TLSv1.2 and TLSv1.3 when acting as a TLS client to provide a trusted channel to the Syslog server. The Cube will always initiate the connection using one of the following ciphersuites:

TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384

TLS 1.3: TLS_AES_256_GCM_SHA384

This ciphersuite can be configured as described in sections 3.8 and 3.10.

When the TOE acts as a TLS/HTTPS server to provide the administrative web GUI, TLSv1.2 and TLSv1.3 are supported and use the following ciphersuites:

TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384

TLS 1.3: TLS_AES_256_GCM_SHA384

To modify the logging parameters of the TOE, log into the GSS GUI and navigate to the following:

'System Settings > Logging'

and scroll down to each of the following settings:

- **'Enable log shipping' – Must be enabled.**
- **'Log destination host' – Must be configured.**
- **'Log shipping port' – Must be configured.**
- **'Log shipping protocol' – Must be set to TCP.**

o 'Log shipping TLS cipher' – Default is TLS 1.2: AES128_GCM_SHA256 **Must be configured to one of:** **TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384** **TLS 1.3: TLS_AES_256_GCM_SHA384** **Each Cube will transmit audit log traffic via TLS through the established IPsec tunnel. The IPSec header is then stripped by GSS and traffic is processed by the firewall ruleset which forwards syslog traffic to the syslog server. GSS initiates the connection to the syslog server at startup, and the Cube will initiate a connection to the syslog server once the IPSec tunnel has been successfully established**

Verdict:

PASS.

4.1.3.3 FCS_CKM.4 CRYPTOGRAPHIC KEY DESTRUCTION (CPP_ND_V3.0E)

4.1.3.3.1 FCS_CKM.4 TSS

The evaluator shall examine 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 shall confirm 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). In particular, if a TOE claims not to store plaintext keys in non-volatile memory then the evaluator shall check that this is consistent with the operation of the TOE.

Evaluator Findings:
The evaluator examined 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. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.2 labeled FCS_CKM.4.** Upon investigation, the evaluator found that the TSS states that: **ECDSA private key – stored on the hard drive and overwritten when a factory reset is performed** **IPsec session key – stored in volatile memory and overwritten when a session is terminated** **TLS session key - stored in volatile memory and overwritten when a session is terminated** **Administrator passwords – Plaintext value is stored in volatile memory when supplied by a user and overwritten after validation; configured administrator passwords are stored on the hard drive as hashed (SHA-256) values only.**

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).

Evaluator Findings:
The evaluator confirmed 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). In particular, TOE Summary Specification section 6.3.2 labeled FCS_CKM.4.the evaluator checked that the claim not to store plaintext keys in non-volatile memory is consistent with the operation of the TOE. The relevant information is found in the following section(s): TOE Summary Specification TOE Summary Specification section 6.3.2 labeled FCS_CKM.4. Upon investigation, the evaluator found that the TSS states that: **The TOE leverages the underlying filesystem to facilitate the destruction of cryptographic keys. Cryptographic keys in volatile and non-volatile memory are overwritten with zeroes when a key is deleted. Keys in volatile memory are also destroyed when a component is powered down or rebooted.**

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 nonvolatile 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.

Evaluator Findings:
N/A/. This activity is not applicable because another selection was made.

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.

Evaluator Findings:

Where the TSS identifies keys that are stored in a non-plaintext form, the evaluator checked 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 relevant information is found in the following section(s): **Table 17: Key Zeroization**

Upon investigation, the evaluator found that the TSS states that:

Keys/CSPs	Storage Location	How Key is Protected	How Key is Derived	When Key is Destroyed	Method of Zeroization
Authentication Keys used for IPsec	Protected file	AES-256 encrypted	KeyGen	X.509 cert deleted	Overwrite with zeroes
PPKs	Protected file	AES-256 encrypted	Imported	PPK deleted	Overwrite with zeroes
Passwords	Protected file	SHA-256 hash only	SHA-256 hash of admin input	Account deleted	Overwrite with zeroes

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.

Evaluator Findings:

The evaluator checked 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).

The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification section 6.3.2 labeled FCS_CKM.4.**

Upon investigation, the evaluator found that the TSS states that: **No configurations or circumstances exist that do not conform to the key destruction requirement.**

Where the ST specifies the use of "a value that does not contain any CSP" to overwrite keys, the evaluator shall examine 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.

Evaluator Findings:

N/A. The TOE overwrites data with zeroes therefore does not use CSPs.

Verdict:

PASS.

4.1.3.3.2 FCS_CKM.4 AGD

A TOE may be subject to situations that could prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS (and any

other supporting information used). The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer. For example, when the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-levelling and garbage collection. This may result in additional copies of the key that are logically inaccessible but persist physically. Where available, the TOE might then describe use of the TRIM command[3] and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

Evaluator Findings:
N/A. The TOE is not subject to any situations that prevent or delay key destruction.

Verdict:

PASS.

4.1.3.4 FCS_COP.1/DATAENCRYPTION CRYPTOGRAPHIC OPERATION (AES DATA ENCRYPTION/DECRYPTION) (CPP_ND_V3.0E)

4.1.3.4.1 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.

Evaluator Findings:
The evaluator examined the TSS to ensure it identifies the key size(s) and mode(s) supported by the TOE for data encryption/decryption. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.3 labeled "FCS_COP.1/DataEncryption"** Upon investigation, the evaluator found that the TSS states that: **The TOE performs data encryption and decryption functions using AES 256 bit encryption in GCM mode as follows:** • **TLS Client to syslog server – AES-GCM, 256 bit keys** • **TLS Server for administrative GUI - AES-GCM, 256 bit keys** • **IKE/IPsec –AES-GCM, 256 bit keys**

Verdict:

PASS.

4.1.3.4.2 FCS_COP.1/DATAENCRYPTION AGD

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.

Evaluator Findings:

The evaluator verified 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.

The relevant information is found in the following section(s): **"VPN"**, **"Cryptographic Function" and "Audit Logging"**

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

To modify the VPN parameters of the TOE, log into the GSS GUI as described in section Error! Reference source not found. **and navigate to the following:**

'*System Settings > VPN*'

and scroll down to each of the following settings:

- **'*IPSec Protocol Mode*' – Set to "IKEv2_Certificates"**
- **'*IPSec Cipher Suite*' – Default is aes-256gcm16-pfsha512-ecp521 (Top Secret DH21)**

Can configure to:

aes-256gcm16-pfsha384-ecp384 (Top Secret DH20)

To modify the logging parameters of the TOE, log into the GSS GUI and navigate to the following:

'System Settings > Logging'

and scroll down to each of the following settings:

- **'*Enable log shipping*' – Must be enabled.**
- **'*Log destination host*' – Must be configured.**
- **'*Log shipping port*' – Must be configured.**
- **'*Log shipping protocol*' – Must be set to TCP.**
- **'*Log shipping TLS cipher*' – Default is TLS 1.2: AES128-GCM-SHA256**

  Must be configured to one of:

  TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384

  TLS 1.3: TLS_AES_256_GCM_SHA384

The TOE only supports TLSv1.2 and TLSv1.3 when acting as a TLS client to provide a trusted channel to the Syslog server. The Cube will always initiate the connection using one of the following ciphersuites:

TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384

TLS 1.3: TLS_AES_256_GCM_SHA384

This ciphersuite can be configured as described in sections "VPN" and "Audit Logging".

When the TOE acts as a TLS/HTTPS server to provide the administrative web GUI, TLSv1.2 and TLSv1.3 are supported and use the following ciphersuites:

TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384

TLS 1.3: TLS_AES_256_GCM_SHA384

Verdict:

PASS.

4.1.3.5 FCS_COP.1/SIGGEN CRYPTOGRAPHIC OPERATION (SIGNATURE GENERATION AND VERIFICATION) (CPP_ND_V3.0E)

4.1.3.5.1 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.

Evaluator Findings:
The evaluator examined the TSS to determine that it specifies the cryptographic algorithm and key size supported by the TOE for signature services. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.4 labeled "FCS_COP.1/SigGen"** Upon investigation, the evaluator found that the TSS states that: **The TOE provides cryptographic signature generation and verification services using:** **ECDSA P-384 and P-521 SigGen to support IPsec and TLS functions.** **ECDSA P-384 and P-521 SigVer to support IPsec, TLS, X.509, trusted update functions including firmware integrity checking.**

Verdict:

PASS.

4.1.3.5.2 FCS_COP.1/SIGGEN AGD

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.

Evaluator Findings:
The evaluator verified that the AGD guidance instructs the administrator how to configure the TOE to use the selected cryptographic algorithm and key size defined in the Security Target supported by the TOE for signature services. The relevant information is found in the following section(s): **"Cryptographic Function" and "Certificate Management"** Upon investigation, the evaluator found that the AGD activity states that: **"Cryptographic Function" states the TOE performs cryptographic key generation services during IPsec tunnel establishment for IKE peer authentication using ECDSA P-384 and P-521 Elliptic curve keys as specified in FIPS Pub 186-4 "Digital Signature Standard (DSS)" Appendix B.4.** **The TOE performs cryptographic signature generation and verification services using ECDSA P-384 and P-521 in support of IPsec, TLS, X509, and trusted update functions. Additional information on certificate parameters can be found in section 3.12** **"Certificate Management" states that the TOE supports ECDSA key pair generation using P-384 and P-521 curves which is available through the TOE GUI. When generating certificate signing requests, the option is given to display the CSR information on screen to copy or can be downloaded in PEM format. Keys generated during this process are stored on the TOE in non-volatile memory and are not accessible by any interface. Keys are overwritten when a factory reset is performed.** **The TOE supports ECDSA key establishment using P-384 and P-521 curves.** **An external CA certificate and key may be loaded onto the system, or a certificate bundle can be loaded into the TOE via the GUI to provide trust for certificate chains and designate trust anchors:** **On the Cube, navigate to the following:** ***'Device > Certificate Management'*** **Note: During the configuration of the server profile (Virtual Server setup) on the Cube, a drop-down menu is provided to select a certificate to use for the connection.** **On GSS, navigate to the following:** ***'Maintenance > Certificates'*** **Note: When importing a certificate onto GSS, options for "v2" or "WebDashboard" are provided for the administrator to define the certificate usage.** **Certificate signing requests can also be generated through the GUI:** **On the Cube, navigate to the following:** ***'Device > Certificate Management > Manage > Create CSR'*** **Key size P-384 or P-521 can be selected from the drop-down menu.** **On GSS, navigate to the following:** ***'Maintenance > Certificates > Create CSR'*** **Key size P-384 or P-521 can be selected from the drop-down menu.**

Verdict:

PASS.

4.1.3.6 FCS_COP.1/HASH CRYPTOGRAPHIC OPERATION (HASH ALGORITHM) (CPP_ND_V3.0E)

4.1.3.6.1 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.

Evaluator Findings:
The evaluator checked that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.5 labeled "FCS_COP.1/Hash"** Upon investigation, the evaluator found that the TSS states that: **The TOE provides cryptographic hashing services using SHA-256, SHA-384 and SHA-512.** **SHA-384 and SHA-512 are used for SigGen and SigVer operations; SHA-256 is only used for hashing configured passwords. The hash algorithms are also used in the associated HMAC algorithms.** **SHA-384 and SHA-512 are used with TLS client connections to the syslog server.** **SHA-384 and SHA-512 are used with IKE/IPsec connections.** **SHA-384 and SHA-512 is used for TLS server connections for the administrative web GUI.**

Verdict:

PASS.

4.1.3.6.2 FCS_COP.1/HASH AGD

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

Evaluator Findings:
The evaluator checked the AGD documents to determine that any configuration that is required to configure the required hash sizes is present. The relevant information is found in the following section(s): **"VPN" and "Cryptographic Functions"** Upon investigation, the evaluator found that the AGD activity states that: **'IPSec Cipher Suite' – Default is aes-256gcm16-pfsha512-ecp521 (Top Secret DH21)** **Can configure to:** **aes-256gcm16-pfsha384-ecp384 (Top Secret DH20)** **The TOE only supports TLSv1.2 and TLSv1.3 when acting as a TLS client to provide a trusted channel to the Syslog server. The Cube will always initiate the connection using one of the following ciphersuites:** **TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384** **TLS 1.3: TLS_AES_256_GCM_SHA384** **This ciphersuite can be configured as described in sections 3.8 and 3.10.** **When the TOE acts as a TLS/HTTPS server to provide the administrative web GUI, TLSv1.2 and TLSv1.3 are supported and use the following ciphersuites:** **TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384** **TLS 1.3: TLS_AES_256_GCM_SHA384**

Verdict:

PASS.

4.1.3.7 FCS_COP.1/KEYEDHASH CRYPTOGRAPHIC OPERATION (KEYED HASH ALGORITHM) (CPP_ND_V3.0E)

4.1.3.7.1 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.

Evaluator Findings:
The evaluator examined 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. The relevant information is found in the following section(s): **TOE Summary Specification section labeled "FCS_COP.1/KeyedHash"** Upon investigation, the evaluator found that the TSS states that: **The TOE provides keyed-hashing message authentication services using HMAC-SHA-384 and HMAC-SHA-512.** **TLS client connections to the syslog server use key sizes [384] bits, [1024] bit block size, and [384] bit message digest size. The TOE uses HMAC-SHA-384 and HMAC-SHA-512 TLS Key Derivation Function (KDF) and TLS message authentication, and the administrative GUI uses HMAC-SHA-384 and HMAC-SHA-512 with key sizes [384, 512] bits, [1024] bit block size, and [384, 512] bit message digest size.** **The relevant NIST CAVP certificate numbers are listed in** Error! Reference source not found. **of the ST.**

Verdict:

PASS.

4.1.3.7.2 FCS_COP.1/KEYEDHASH AGD

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.

Evaluator Findings:
The evaluator verified 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. The relevant information is found in the following section(s): **"Cryptographic Function"** Upon investigation, the evaluator found that the AGD activity states that: **Keyed hash functions are not configurable on their own and are instead included with the overall selection of the desired ciphersuite. For example, ciphersuites using SHA384 will include HMAC-SHA-384.**

Verdict:

PASS.

4.1.3.8 FCS_HTTPS_EXT.1 HTTPS PROTOCOL (CPP_ND_V3.0E)

4.1.3.8.1 FCS_HTTPS_EXT.1 TSS

The evaluator shall examine the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818.

Evaluator Findings:
The evaluator examined the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818.

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

Upon investigation, the evaluator found that the TSS states that: **The TOE acts as a TLS/HTTPS server in accordance with RFC 2818 to provide a web GUI to administrators. All MUST and REQUIRED requirements specified in RFC 2818 are followed.**

Verdict:

PASS.

4.1.3.8.2 FCS_HTTPS_EXT.1 AGD

The evaluator shall examine the guidance documentation to verify it instructs the Administrator how to configure TOE for use as an HTTPS client or HTTPS server.

Evaluator Findings:

The evaluator examined the guidance documentation to verify it instructs the Administrator how to configure TOE for use as an HTTPS client or HTTPS server.

The relevant information is found in the following section(s): **"Administrative Interfaces"**

Upon investigation, the evaluator found that the AGD activity states that: **Only the following administration interfaces may be used:**

- **GoSilent Server GUI. Administrative GUI interface for managing all GoSilent configurations on the server and other GoSilent platforms.**
  - **Local access is provided for the configured whitelisted source IP address of an administrator device, and by navigating to the local IP address of GSS through the (virtual) MGMT Ethernet interface using HTTPS via web browser to TCP port 4439.**
  - **Remote access is provided to administrators connecting from Cube clients (for any client that has a valid GSS administrator account) using HTTPS via web browser to the MGMT IP address of the GSS over TCP port 443.**
- **The GoSilent Cube GUI user interface provides limited functionality for initial Cube configuration.**
  - **Local access is provided to users by navigating to the local IP address of the cube using HTTPS via web browser over port 4439 or by navigating to the URL** https://setup.gosilent.
  - **Configuration of some Cube functions is done via the GSS GUI. The Cube subsequently downloads these configuration settings from GSS upon connection establishment.**

Verdict:

PASS.

4.1.3.9 FCS_IPSEC_EXT.1 IPsec Protocol (CPP_ND_V3.0E)

4.1.3.9.1 FCS_IPSEC_EXT.1.1 TSS

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.

Evaluator Findings:
The evaluator examined 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. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **The GoSilent Server has a nominal Security Policy Database (SPD) that denies all traffic. When an IPsec tunnel to GoSilent Server is established, an SPD entry is dynamically added that permits all traffic received from user systems with an IP source address that is consistent with the IP subnet address configured for the GoSilent Cube IPsec interface. No traffic from a user system is forwarded to the TOE Network other than through an IPsec tunnel (to GoSilent Server).** **The GoSilent Server has a nominal SPD that denies all traffic to any GoSilent Cube. When an IPsec tunnel from each GoSilent Cube is established, an SPD entry is dynamically added that permits all traffic destined for an IP address that is consistent with the IP subnet address configured for the GoSilent Cube IPsec interface.**

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.

Evaluator Findings:
As noted in Section 4.4.1 of RFC 4301, the processing of entries in the SPD is non-trivial and the evaluator determined 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 determined 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 covers 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. The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **The GoSilent Server has a nominal SPD that denies all traffic to any GoSilent Cube. When an IPsec tunnel from each GoSilent Cube is established, an SPD entry is dynamically added that permits all traffic destined for an IP address that is consistent with the IP subnet address configured for the GoSilent Cube IPsec interface.**

Verdict:

PASS.

4.1.3.9.2 FCS_IPSEC_EXT.1.2 TSS

None.

4.1.3.9.3 FCS_IPSEC_EXT.1.3 TSS

The evaluator shall check 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).

Evaluator Findings:
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). The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **The TOE implements the IPsec architecture per RFC 4301 using tunnel mode.**

Verdict:

PASS.

4.1.3.9.4 FCS_IPSEC_EXT.1.4 TSS

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.

Evaluator Findings:
The evaluator examined 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. The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **IPsec support includes:** **Hash functions as specified in RFC 4868**

Verdict:

PASS.

4.1.3.9.5 FCS_IPSEC_EXT.1.5 TSS

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

Evaluator Findings:
The evaluator examined the TSS to verify that IKEv1 and/or IKEv2 are implemented. The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Only IKEv2 is supported.**

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.

Evaluator Findings:
N/A. The TOE only supports IKEv2.

Verdict:

PASS.

4.1.3.9.6 FCS_IPSEC_EXT.1.6 TSS

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.

Evaluator Findings:
The evaluator ensured 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. The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Only IKEv2 is supported. AES in GCM mode using 256-bit keys is supported for both IKEv2 and ESP.**

Verdict:

PASS.

4.1.3.9.7 FCS_IPSEC_EXT.1.7 TSS

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.

Evaluator Findings:
The evaluator ensured the TSS identifies the lifetime configuration method used for limiting the IKEv1 Phase 1 SA lifetime and/or the IKEv2 SA lifetime. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Only IKEv2 is supported. SA lifetimes based on time, configurable between 1 and 24 hours.**

The evaluator shall verify that the selection made here corresponds to the selection in FCS_IPSEC_EXT.1.5.

Evaluator Findings:
The evaluator verified that the selection made here corresponds to the selection in FCS_IPSEC_EXT.1.5. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **IPsec support includes:** • **NAT traversal as specified in RFC 7296 section 2.23** • **Hash functions as specified in RFC 4868**

Verdict:

PASS.

4.1.3.9.8 FCS_IPSEC_EXT.1.8 TSS

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.

Evaluator Findings:
The evaluator ensured the TSS identifies the lifetime configuration method used for limiting the IKEv1 Phase 2 SA lifetime and/or the IKEv2 Child SA lifetime. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Only IKEv2 is supported.** **Child SA lifetimes based on time, configurable between 1 and 8 hours.**

The evaluator shall verify that the selection made here corresponds to the selection in FCS_IPSEC_EXT.1.5.

Evaluator Findings:
The evaluator verified that the selection made here corresponds to the selection in FCS_IPSEC_EXT.1.5. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **IPsec support includes:** • **NAT traversal as specified in RFC 7296 section 2.23** • **Hash functions as specified in RFC 4868**

Verdict:

PASS.

4.1.3.9.9 FCS_IPSEC_EXT.1.9 TSS

The evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating "x".

Evaluator Findings:
The evaluator checked and ensured that for each DH group supported, the TSS describes the process for generating "x". The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **DH groups 19 and 20 (security strengths 192 and 256 bits) are used.** **Size of the random secret value "x" and nonce used for key establishment, as generated by the DRBG, is at least twice the security strength of that associated with the negotiated DH group.**

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.

Evaluator Findings:
The evaluator verified 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. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Size of the random secret value "x" and nonce used for key establishment, as generated by the DRBG, is at least twice the security strength of that associated with the negotiated DH group.**

Verdict:
PASS.

4.1.3.9.10 FCS_IPSEC_EXT.1.10 TSS

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.

Evaluator Findings:

If the first selection is chosen, the evaluator checked and ensured that for each DH group supported, the TSS describes the process for generating each nonce.
The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"**
Upon investigation, the evaluator found that the TSS states that: **Size of the random secret value "x" and nonce used for key establishment, as generated by the DRBG, is at least twice the security strength of that associated with the negotiated DH group. The nonces used in IKE exchanges are generated in a manner such that the probability that a specific nonce value will be repeated during the life of a specific IPsec SA is less than 1 in 2^[192]. Nonce lengths are dynamically determined based on the negotiated Diffie-Hellman group or the pseudorandom function (PRF) hash as follows:**
• **Nonce lengths must be sufficiently large to support all TOE-chosen proposals during the exchange, as nonces may be exchanged before the Diffie-Hellman group is negotiated.**
• **For Diffie-Hellman Groups 20 and 21, the nonce length is at least 192 bits.**
• **When the negotiated PRF hash is used to determine nonce length, the nonce is set to at least half the output size of the PRF hash:**
  ○ **For HMAC-SHA-384, the nonce length is at least 192 bits.**
  ○ **For HMAC-SHA-512, the nonce length is at least 256 bits.**

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.

Evaluator Findings:
The evaluator verified 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. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **size of the random secret value "x" and nonce used for key establishment, as generated by the DRBG, is at least twice the security strength of that associated with the negotiated DH group.**

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.

Evaluator Findings:
If the second selection is chosen, the evaluator checked and ensured that for each PRF hash supported, the TSS describes the process for generating each nonce. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Selection not chosen. This activity is considered satisfied.**

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.

Evaluator Findings:
Selection not chosen, this activity is considered not applicable.

Verdict:

PASS.

4.1.3.9.11 FCS_IPSEC_EXT.1.11 TSS

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.

Evaluator Findings:
The evaluator checked 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. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that**: Diffie–Hellman (DH) groups 20 and 21 (security strengths 192 and 256 bits)**

Verdict:

PASS.

4.1.3.9.12 FCS_IPSEC_EXT.1.12 TSS

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 and ensured 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.

Evaluator Findings:
The evaluator checked 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 also describes the checks that are done when negotiating IKEv1 Phase 2 and/or IKEv2 CHILD_SA suites and ensured 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. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Only IKEv2 is supported. AES in GCM mode using 256-bit keys is supported for both IKEv2 and ESP.**

Verdict:

PASS.

4.1.3.9.13 FCS_IPSEC_EXT.1.13 TSS

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).

Evaluator Findings:
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). The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Peer authentication using ECDSA certificates with key sizes 384 or 512 bits**.

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.

Evaluator Findings:
If pre-shared keys are chosen in the selection, the evaluator checked and ensured that the TSS describes how pre-shared keys are established and used in authentication of IPsec connections. The description in the TSS also indicates 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 relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **The TOE supports RFC 8784 PPKs with IPsec. Administrators may import (load) externally-generated bit-based PPKs and their associated PPK-IDs onto GoSilent Server and GoSilent Cube, and then configure a PPK-ID to be used with a specific remote endpoint. If an endpoint is configured for a connection, then PPKs must be used for that connection and the same PPK-ID must be configured for both GoSilent Server and GoSilent Cube. If no PPK-ID is configured, standard IKEv2 protocols are used for the connection.**

Verdict:

PASS.

4.1.3.9.14 FCS_IPSEC_EXT.1.14 TSS

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.

Evaluator Findings:
The evaluator ensured that the TSS describes how the TOE compares the peer's presented identifier to the reference identifier. This description includes 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 describes 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 states this. If the ST author assigned an additional identifier type, the TSS description also includes 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. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.7 labeled "FCS_IPSEC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Peer authentication using ECDSA certificates with key sizes 384 or 512 bits. Distinguished Names as reference identifiers.**

Verdict:

PASS.

4.1.3.9.15 FCS_IPSEC_EXT.1.1 AGD

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.

Evaluator Findings:
The evaluator examined 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 relevant information is found in the following section(s): **"Firewall Configuration"** Upon investigation, the evaluator found that the AGD section labelled **"Firewall Configuration states that packet filtering rules can be configured with an action to accept (permit/allow), reject (discard/deny), ignore, or to pass the packet on to other rules for more processing. The rest of section 3.3 provides instructions on packet related configuration.**

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.

Evaluator Findings:

The evaluator determined 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.
The relevant information is found in the following section(s): **"Firewall Configuration"**
 	
Upon investigation, the evaluator found that the AGD activity states that: **Packet filtering rules can be configured with an action to accept (permit/allow), reject (discard/deny), ignore, or to pass the packet on to other rules for more processing.**
Netfilter allows the definition of packet filtering rules and processes incoming or outgoing traffic according to the following parameters:
• **IPv4 (RFC 791)**
  • **Source Address**
  • **Destination Address**
  • **Protocol**
• **IPv6 (RFC 8200)**
  • **Source Address**
  • **Destination Address**
  • **Next Header (Protocol)**
• **TCP (RFC 793)**
  • **Source Port**
  • **Destination Port**
• **UDP (RFC 768)**
  • **Source Port**
  • **Destination Port**

Verdict:

PASS.

4.1.3.9.16 FCS_IPSEC_EXT.1.2 AGD

None.

4.1.3.9.17 FCS_IPSEC_EXT.1.3 AGD

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

Evaluator Findings:
The evaluator confirmed that the guidance documentation contains instructions on how to configure the connection in each mode selected.

The relevant information is found in the following section(s): **"VPN"** Upon investigation, the evaluator found that the AGD activity states that: **GoSilent Cubes always initiate the connection to establish an IPsec tunnel with GSS. The TOE only supports tunnel mode; thus no further configuration is needed.**

Verdict:

PASS.

4.1.3.9.18 FCS_IPSEC_EXT.1.4 AGD

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

Evaluator Findings:
The evaluator checked the guidance documentation to ensure it provides instructions on how to configure the TOE to use the algorithms selected. The relevant information is found in the following section(s): **"VPN"** Upon investigation, the evaluator found that the AGD activity states that: **To modify the VPN parameters of the TOE, log into the GSS GUI as described in the "Administrative Interfaces" section and navigate to the following:** **'System Settings > VPN'** **and scroll down to each of the following settings:** **o 'IPSec Protocol Mode' – Set to "IKEv2_Certificates"** **o 'IPSec Cipher Suite' – Default is aes-256gcm16-pfsha512-ecp521 (Top Secret DH21)** **• Can configure to:** **aes-256gcm16-pfsha384-ecp384 (Top Secret DH20)**

Verdict:

PASS.

4.1.3.9.19 FCS_IPSEC_EXT.1.5 AGD

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).

Evaluator Findings:
The evaluator checked the AGD 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). The relevant information is found in the following section(s): **"VPN"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE only supports IKEv2. This is made clear in the "VPN" section to configure the TOE to use 'IKEv2_Certificates' for 'IPsec Protocol Mode'.**

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.

Evaluator Findings:
N/A – The TOE does not support IKEv1.

Verdict:

PASS.

4.1.3.9.20 FCS_IPSEC_EXT.1.6 AGD

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

Evaluator Findings:
The evaluator ensured that the guidance documentation describes the configuration of all selected algorithms in the requirement. The relevant information is found in the following section(s): **"VPN"** Upon investigation, the evaluator found that the AGD activity states that: **To modify the VPN parameters of the TOE, log into the GSS GUI as described in the "Administrative Interfaces" section and navigate to the following:** **'System Settings > VPN'** **and scroll down to each of the following settings:** **o 'IPSec Protocol Mode' – Set to "IKEv2_Certificates"** **o 'IPSec Cipher Suite' – Default is aes-256gcm16-pfsha512-ecp521 (Top Secret DH21)** **Can configure to:** **o aes-256gcm16-pfsha384-ecp384 (Top Secret DH20)**

Verdict:

PASS.

4.1.3.9.21 FCS_IPSEC_EXT.1.7 AGD

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).

Evaluator Findings:
The evaluator verified 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 relevant information is found in the following section(s): **"VPN"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE only supports time-based limits. To modify the VPN parameters of the TOE, log into the GSS GUI as described in section 3.1 and navigate to the following:** **'System Settings > VPN'** **and scroll down to each of the following settings:** **o 'Security Association (SA) Lifetime' – Between 1 and 24 hours** **o 'Child SA Lifetime' – Between 1 and 8 hours**

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.

Evaluator Findings:
The evaluator verified 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. The relevant information is found in the following section(s): **"VPN"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE only supports time-based limits. To modify the VPN parameters of the TOE, log into the GSS GUI as described in section 3.1 and navigate to the following:** **'System Settings > VPN'** **and scroll down to each of the following settings:** **o 'Security Association (SA) Lifetime' – Between 1 and 24 hours** **o 'Child SA Lifetime' – Between 1 and 8 hours**

Verdict:

PASS.

4.1.3.9.22 FCS_IPSEC_EXT.1.8 AGD

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).

Evaluator Findings:
The evaluator verified 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 relevant information is found in the following section(s): **"VPN"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE only supports time-based limits. To modify the VPN parameters of the TOE, log into the GSS GUI as described in section 3.1 and navigate to the following:** **'System Settings > VPN'** **and scroll down to each of the following settings:** **o 'Security Association (SA) Lifetime' – Between 1 and 24 hours** **o 'Child SA Lifetime' – Between 1 and 8 hours**

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.

Evaluator Findings:
The evaluator verified 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. The relevant information is found in the following section(s): **"VPN"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE only supports time-based limits. To modify the VPN parameters of the TOE, log into the GSS GUI as described in section 3.1 and navigate to the following:** **'System Settings > VPN'** **and scroll down to each of the following settings:** **o 'Security Association (SA) Lifetime' – Between 1 and 24 hours** **o 'Child SA Lifetime' – Between 1 and 8 hours**

Verdict:

PASS.

4.1.3.9.23 FCS_IPSEC_EXT.1.9 AND FCS_IPSEC_EXT.10 AGD

None.

4.1.3.9.24 FCS_IPSEC_EXT.1.11 AGD

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

Evaluator Findings:
The evaluator ensured that the guidance documentation describes the configuration of all algorithms selected in the requirement. The relevant information is found in the following section(s): **"VPN"** Upon investigation, the evaluator found that the AGD activity states that: **To modify the VPN parameters of the TOE, log into the GSS GUI as described in the "Administrative Interfaces" section and navigate to the following:** **'System Settings > VPN'** **and scroll down to each of the following settings:** **o 'IPSec Protocol Mode' – Set to "IKEv2_Certificates"** **o 'IPSec Cipher Suite' – Default is aes-256gcm16-pfsha512-ecp521 (Top Secret DH21)** **• Can configure to:** **o aes-256gcm16-pfsha384-ecp384 (Top Secret DH20)**

Verdict:
PASS.

4.1.3.9.25 FCS_IPSEC_EXT.1.12 AGD

None.

4.1.3.9.26 FCS_IPSEC_EXT.1.13 AGD

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.

Evaluator Findings:
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 relevant information is found in the following section(s): **"Certificate Management"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE supports ECDSA key pair generation using P-384 and P-521 curves which is available through the TOE GUI. When generating certificate signing requests, the option is given to display the CSR information on screen to copy or can be downloaded in PEM format. Keys generated during this process are stored on the TOE in non-volatile memory and are not accessible by any interface. Keys are overwritten when a factory reset is performed.** **The TOE supports ECDSA key establishment using P-384 and P-521 curves.**

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.

Evaluator Findings:
The evaluator checked that the guidance documentation describes how pre- shared keys are to be generated and established. The description in the guidance documentation also indicates 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 relevant information is found in the following section(s): **"Post-quantum Pre-shared Keys Configuration"** Upon investigation, the evaluator found that the AGD activity states that: **PPKs are generated on a third-party system.** **To load a PPK on the GSS, PPK usage must first be enabled. Log into the GUI and navigate to the following:** **'System Settings'** **Scroll down to the following setting:** **o 'PPK Secret Size' – The default value of "256-bit (Standard)" must always be used.** **o 'Enable PPK support for IKEv2 clients' – Must be enabled.** **This step only needs to be performed once, and the GSS must be restarted after enabling the parameter for it to take effect.**

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".

Evaluator Findings:
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". The relevant information is found in the following section(s): **"Certificate Management"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE performs several certificate validity checks on upload, and during TLS and IPsec connection establishment. These checks include a check against its own trust store, the expiration date, revocation status via OCSP (IPSec) and CRL (TLS) and verifies the correct extendedKeyUsage purpose set (Server Authentication). During a certificate upload, the TOE also verifies that the certificate chain terminates with a trusted CA certificate. When validating a certificate used for IPsec, the reference identifier will be matched against the DN (See below for configuration). Certificates used for TLS connections will fall back to the CN if the SAN is not present.**

Verdict:

PASS.

4.1.3.9.27 FCS_IPSEC_EXT.1.14 AGD

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.

Evaluator Findings:
The evaluator ensured 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 ensured that the operational guidance provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.

The relevant information is found in the following section(s): **"Certificate Management"**

Upon investigation, the evaluator found that the AGD activity states that: **When validating a certificate used for IPsec, the reference identifier will be matched against the DN (See below for configuration). Certificates used for TLS connections will fall back to the CN if the SAN is not present.**

Administrators may configure both CN and O fields for values that are expected to be present in certificates received from the remote side during IPsec connection setups. The CN value is configurable by default in the 'System Settings > VPN' menu, however the O value configurability must be enabled using the following steps:

(1) Enable the O value configuration functionality

Log into GSS as an administrator and navigate to 'System Settings > VPN GUI'.

The "Enable Organization for certificate subject name" check box must be selected.

(2) Restart GSS & Cubes

The GSS must be restarted for a change in this setting to take effect. Each Cube must be restarted three (3) times for a change in this setting to take effect:

(i) Triggers updated settings to be downloaded from GSS

(ii) Cube detects the setting change and reconfigure the network functionality.

(iii) Reconfigured networking functionality to take effect on the Cube.

(3) Configure O value for each applicable Cube on GSS

Log in to the GSS as an administrator and navigate to 'System Settings > VPN Clients GUI'. Add a new cube, or edit an existing Cube entry, and configure the Organization value.

(4) Configure the O value for the GSS on each applicable Cube.

Log in to the Cube as an administrator and select the 'Server Profiles' page. Add a new profile or edit an existing profile entry and configure the Organization value.

Note: When generating a CSR, the TOE does not include the 'O' value. This must be manually configured per the instructions above, or is the responsibility of the signing CA to insert the 'O' value into the certificate.

The TOE will check the 'O' value if it is present in the certificate.

Verdict:

PASS.

4.1.3.10 FCS_IPSEC_EXT.1/VPN IPSEC PROTOCOL (MOD_VPNGW_V1.3)

4.1.3.10.1 FCS_IPSEC_EXT.1/VPN TSS (MOD_VPNGW_V1.3)

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

Evaluator Findings:

The evaluator reviewed the TSS, section 6.10.1 "FTP_ITC.1, FTP_ITC.1/VPN" and ensured that it describes how the selection works in conjunction with the authentication of IPsec connections.

The TOE supports RFC 8784 PPKs with IPsec. Administrators may import (load) externally-generated bit-based PPKs and their associated PPK-IDs onto GoSilent Server and GoSilent Cube, and then configure a PPK-ID to be used with a specific remote endpoint. If an endpoint is configured for a connection, then PPKs must be used for

that connection and the same PPK-ID must be configured for both GoSilent Server and GoSilent Cube. If no PPK-ID is configured, standard IKEv2 protocols are used for the connection

Verdict:
PASS.

4.1.3.10.2 FCS_IPSEC_EXT.1 AGD/VPN (MOD_VPNGW_V1.3)

If any selection with "Pre-shared Keys" is selected, the evaluator shall check that the operational guidance describes any configuration necessary to enable any selected authentication mechanisms.

Evaluator Findings:
The evaluator checked the AGD section labeled "**Post-quantum Pre-Shared Keys Configuration**" and ensured that it describes any configuration necessary to enable any selected authentication mechanisms. **Post-quantum Pre-shared Keys (PPKs) as defined in RFC 8784 may be used as an enhancement to the standard ECDHE key establishment process. A PPK is an additional secret shared by IPsec endpoints that is stirred into the standard critical security parameters used during key establishment, which provides quantum resistance to the resulting Security Associations (SAs).** **PPKs are generated on a third-party system. Since PPK values are secret, they are never displayed after being loaded into GoSilent or directly exchanged between the endpoints during IPsec connection establishment. Instead, PPKs are referenced via PPK_ID values, which are also generated on the third-party system. A {PPK_ID, PPK} pair is bound together when the values are generated by the third-party system and communicated out of band to authorized administrators. PPKs must be unique, and the PPK_IDs must also be unique. The pair is loaded into the GSS and GSC via the GUI by an authorized administrator. Once loaded, further management of each pair is performed using the PPK_ID only.** **PPK_IDs are configured on a per-connection basis. On the GSS, each PPK_ID may be configured for just one VPN Client (e.g. Cube). On the Cube, a single PPK_ID may be configured in more than one profile (e.g. GSS). Since a Cube only has a single active IPsec connection at any time, and any PPK_ID can only be configured for one Cube on each GSS, this scenario ensures that each PPK can only be in use with one IPsec connection at any time.**

Verdict:
PASS.

4.1.3.11 FCS_RBG_EXT.1 RANDOM BIT GENERATION (CPP_ND_V3.0E)

4.1.3.11.1 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.

Evaluator Findings:
The evaluator examined the TSS and determined 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. The relevant information is found in the following section(s): **TOE Summary Specification section labeled "FCS_RBG_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **The TOE implements a Counter DRBG (AES), compliant with ISO/IEC 18031:2011 (SP800-90B), to generate random bits needed for asymmetric key, symmetric key, nonce, and salt generation. The DRBG is seeded with 256 bits of entropy from one software-based noise source.** **Additional detail is provided in the proprietary Entropy Description.**

Verdict:

PASS.

4.1.3.11.2 FCS_RBG_EXT.1 AGD

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

Evaluator Findings:
The evaluator confirmed that the guidance documentation contains appropriate instructions for configuring the RNG functionality. The relevant information is found in the following section(s): **"Cryptographic Function"** Upon investigation, the evaluator found that the AGD activity states that: **The RNG functionality provided by the TOE does not require additional configuration.**

Verdict:

PASS.

4.1.3.12 FCS_TLSC_EXT.1 TLS CLIENT PROTOCOL (CPP_ND_V3.0E)

4.1.3.12.1 FCS_TLSC_EXT.1.1 TSS

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

Evaluator Findings:
The evaluator checked the description of the implementation of this protocol in the TSS and ensured that the ciphersuites supported are specified. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.9 labeled "FCS_TLSC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Both the GoSilent Server and GoSilent Cube act as a TLS client to provide a trusted channel to the syslog server. The TOE initiates this connection. This client (on both TOE components) supports TLSv1.2 and TLSv1.3 with the following configurable ciphersuites:** • **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384** • **TLS_AES_256_GCM_SHA384**

Verdict:

PASS.

4.1.3.12.2 FCS_TLSC_EXT.1.2 TSS

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

Evaluator Findings:
The evaluator ensured that the TSS describes the client's method of establishing all reference identifiers from the administrator/application- configured reference identifier, including which types of reference identifiers are supported (e.g. application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.9 labeled "FCS_TLSC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **The TOE automatically parses the reference identifier from the connection parameters, using the hostname or IP address as the reference identifier. When validating the server certificate, the TSF matches the reference identifier against the SAN:DNS or SAN:IP address field in the presented certificate (if present) per RFC 6125 section 6 and falls back to the CN if the SAN is not present. The TOE treats the CN as a string value and compares the configured syslog server string to that value. The TOE does not support wildcards.**

Note that where a TLS channel is being used between components of a distributed TOE for FPT_ITT.1, the requirements to have the reference identifier established by the administrator are relaxed and the identifier may also be established through a "Gatekeeper" discovery process. The TSS shall describe the discovery process and highlight how the reference identifier is supplied to the "joining" component. Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1 and the ST author selected attributes from RFC 5280, the evaluator shall ensure the TSS describes which attribute type, or combination of attributes types, are used by the client to match the presented identifier with the configured identifier. The evaluator shall ensure the TSS

presents an argument how the attribute type, or combination of attribute types, uniquely identify the remote TOE component; and the evaluator shall verify the attribute type, or combination of attribute types, is sufficient to support unique identification of the maximum supported number of TOE components.

Evaluator Findings:
Note that where a TLS channel is being used between components of a distributed TOE for FPT_ITT.1, the requirements to have the reference identifier established by the administrator are relaxed and the identifier may also be established through a "Gatekeeper" discovery process. The TSS describes the discovery process and highlights how the reference identifier is supplied to the "joining" component. Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1 and the ST author selected attributes from RFC 5280, the evaluator ensured the TSS describes which attribute type, or combination of attributes types, are used by the client to match the presented identifier with the configured identifier. The evaluator ensured the TSS presents an argument how the attribute type, or combination of attribute types, uniquely identify the remote TOE component; and the evaluator verified the attribute type, or combination of attribute types, is sufficient to support unique identification of the maximum supported number of TOE components. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.9 labeled "FCS_TLSC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **TLS is not a channel used between components. This activity is considered satisfied.**

If IP addresses are supported in the CN as reference identifiers, the evaluator shall ensure that the TSS describes the TOE's conversion of the text representation of the IP address in the CN to a binary representation of the IP address in network byte order.

Evaluator Findings:
If IP addresses are supported in the CN as reference identifiers, the evaluator ensured that the TSS describes the TOE's conversion of the text representation of the IP address in the CN to a binary representation of the IP address in network byte order. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.9 FCS_TLSC_EXT.1.** Upon investigation, the evaluator found that the TSS states that: **The CN string must conform to dotted decimal for IPv4 (RFC 3986) or eight colon separated groups of four lowercase hexadecimal digits for IPv6 (RFC 5952).**

The evaluator shall also ensure that the TSS describes whether canonical format (RFC 5952 for IPv6, RFC 3986 for IPv4) is enforced.

Evaluator Findings:
The evaluator also ensured that the TSS describes whether canonical format (RFC 5952 for IPv6, RFC 3986 for IPv4) is enforced. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.9 FCS_TLSC_EXT.1.** Upon investigation, the evaluator found that the TSS states that: **The CN string must conform to dotted decimal for IPv4 (RFC 3986) or eight colon separated groups of four lowercase hexadecimal digits for IPv6 (RFC 5952).**

Verdict:

PASS.

4.1.3.12.3 FCS_TLSC_EXT.1.3 TSS

None.

4.1.3.12.4 FCS_TLSC_EXT.1.4 TSS

If “present the Supported Groups Extension” is selected, the evaluator shall verify that TSS describes the Supported Groups Extension and whether the required behaviour is performed by default or may be configured. If TLS 1.2 is claimed and DHE ciphers are claimed, then the TSS must also specify whether the TOE is capable of negotiating DHE ciphers and whether the TOE client will terminate if an unsupported DHE parameter set is returned in the Server Key Exchange or whether all valid server-generated DHE parameters are accepted.

Evaluator Findings:
If “present the Supported Groups Extension” is selected, the evaluator verified that TSS describes the Supported Groups Extension and whether the required behaviour is performed by default or may be configured. If TLS 1.2 is claimed and DHE ciphers are claimed, then the TSS also specifies whether the TOE is capable of negotiating DHE ciphers and whether the TOE client will terminate if an unsupported DHE parameter set is returned in the Server Key Exchange or whether all valid server-generated DHE parameters are accepted The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.9 labeled “FCS_TLSC_EXT.1”** Upon investigation, the evaluator found that the TSS states that: **Any SSL/TLS version stated in the Server Hello other than TLS v1.2 or TLSv1.3 causes the connection to be terminated.** **The connection is always terminated if the server certificate is invalid.** **The TOE always presents the Supported Elliptic Curves Extension indicating support for P-384 and P-521 in the Client Hello. The TOE always presents the Signature Algorithms Extension indicating support for P-384 and P-521. Neither Pre-Shared Keys (PSKs) nor secure renegotiation are supported.** **Administrators may select which of the 2 supported ciphersuites are used.**

Verdict:

PASS.

4.1.3.12.5 FCS_TLSC_EXT.1.5 TSS

[Conditional]: The evaluator shall verify that TSS describes the signature_algorithms extension and whether the required behavior is performed by default or may be configured.

Evaluator Findings:
[Conditional]: The evaluator verified that TSS describes the signature_algorithms extension and whether the required behavior is performed by default or may be configured. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.9 labeled "FCS_TLSC_EXT.1"** Upon investigation, the evaluator found that the TSS states that**: The TOE always presents the Supported Elliptic Curves Extension indicating support for P-384 and P-521 in the Client Hello. The TOE always presents the Signature Algorithms Extension indicating support for P-384 and P-521. Neither Pre-Shared Keys (PSKs) nor secure renegotiation are supported.** **Administrators may select which of the 2 supported ciphersuites are used.** **The TSF does not support the use of the following extensions:** • **Early data extension** • **Post-handshake client authentication according to RFC 8446, Section 4.2.6.**

[Conditional]: The evaluator shall verify that TSS describes the signature_algorithms_cert extension and whether the required behavior is performed by default or may be configured.

Evaluator Findings:
The evaluator found that this activity does not apply because the required selection is not applicable.

Verdict:

PASS.

4.1.3.12.6 FCS_TLSC_EXT.1.6 TSS

The evaluator shall verify that TSS describes whether the list of supported ciphersuites can be configured or not.

Evaluator Findings:
The evaluator verified that TSS describes whether the list of supported ciphersuites can be configured or not. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.9 labeled "FCS_TLSC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Administrators may select which of the 2 supported ciphersuites are used.**

Verdict:

PASS.

4.1.3.12.7 FCS_TLSC_EXT.1.7 TSS

None.

4.1.3.12.8 FCS_TLSC_EXT.1.8 TSS

The evaluator shall verify in the TSS that, for TLS 1.3, the TOE shall not permit out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration.

Evaluator Findings:
The evaluator verified in the TSS that, for TLS 1.3, the TOE shall not permit out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.9 labeled "FCS_TLSC_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Neither Pre-Shared Keys (PSKs) nor secure renegotiation are supported.**

Verdict:

PASS.

4.1.3.12.9 FCS_TLSC_EXT.1.9 TSS

None.

4.1.3.12.10 FCS_TLSC_EXT.1.1 AGD

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

Evaluator Findings:
The evaluator checked the guidance documentation and ensured that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS. The relevant information is found in the following section(s): **"Audit Logging" and "Cryptographic Functions"** Upon investigation, the evaluator found that the AGD activity states that: **Each TOE component also transmits a copy of each audit record to an external syslog server in real time via TLS. Syslog server parameters are configured on the GoSilent Server and communicated to all GoSilent Cubes.** **To modify the logging parameters of the TOE, log into the GSS GUI and navigate to the following:** ***'System Settings > Logging'*** **and scroll down to each of the following settings:** ○ **'*Enable log shipping*' – Must be enabled.** ○ **'*Log destination host*' – Must be configured.** ○ **'*Log shipping port*' – Must be configured.** ○ **'*Log shipping protocol*' – Must be set to TCP.** ○ **'*Log shipping TLS cipher*' – Default is TLS 1.2: AES128-GCM-SHA256** **Must be configured to one of:** **TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384** **TLS 1.3: TLS_AES_256_GCM_SHA384**

The TOE only supports TLSv1.2 and TLSv1.3 when acting as a TLS client to provide a trusted channel to the Syslog server. The Cube will always initiate the connection using one of the following ciphersuites: **TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384** **TLS 1.3: TLS_AES_256_GCM_SHA384** **This ciphersuite can be configured as described in sections** Error! Reference source not found. **and** Error! Reference source not found.**.**

Verdict:

PASS.

4.1.3.12.11 FCS_TLSC_EXT.1.2 AGD

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

Evaluator Findings:
The evaluator ensured that the AGD describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE includes support for IP addresses, the evaluator ensured that the AGD provides a set of warnings and/or CA policy recommendations that would result in secure TOE use. The relevant information is found in the following section(s): "**Certificate Management" and "Audit Logging"** Upon investigation, the evaluator found that the AGD activity states that: **Certificates used for TLS connections will fall back to the CN if the SAN is not present.** **To modify the logging parameters of the TOE, log into the GSS GUI and navigate to the following:** **'System Settings > Logging'** **and scroll down to each of the following settings:** **o 'Enable log shipping' – Must be enabled.** **o 'Log destination host' – Must be configured.** **o 'Log shipping port' – Must be configured.** **o 'Log shipping protocol' – Must be set to TCP.** **o 'Log shipping TLS cipher' – Default is TLS 1.2: AES128-GCM-SHA256** **Must be configured to one of:** **TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384** **TLS 1.3: TLS_AES_256_GCM_SHA384** **o 'Reject wildcard TLS server certificates' – Must be enabled.** **o 'Enable CRL checking' – Must be enabled**

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

Evaluator Findings:
N/A – **The TOE does not support TLS for Inter-TOE communication**

Verdict:

PASS.

4.1.3.12.12 FCS_TLSC_EXT.1.3 AGD

None.

4.1.3.12.13 FCS_TLSC_EXT.1.4 AGD

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

Evaluator Findings:
N/A – **The TSS does not indicate that the Supported Elliptic Curves/Supported Groups Extension must be configured to meet the requirement.**

Verdict:

PASS.

4.1.3.12.14 FCS_TLSC_EXT.1.5 AGD

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

Evaluator Findings:
The evaluator verified in the AGD that the guidance includes configuration of the signature_algorithms extension. The relevant information is found in the following section(s): **“Audit Logging”**

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

To modify the logging parameters of the TOE, log into the GSS GUI and navigate to the following:

'System Settings > Logging'

and scroll down to each of the following settings:

- **'Enable log shipping' – Must be enabled.**
- **'Log destination host' – Must be configured.**
- **'Log shipping port' – Must be configured.**
- **'Log shipping protocol' – Must be set to TCP.**
- **'Log shipping TLS cipher' – Default is TLS 1.2: AES128-GCM-SHA256**

Must be configured to one of:

TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384

TLS 1.3: TLS_AES_256_GCM_SHA384

- **'Reject wildcard TLS server certificates' – Must be enabled.**
- **'Enable CRL checking' – Must be enabled.**

The TSS also states: **"The TOE always presents the Supported Elliptic Curves Extension indicating support for P-384 and P-521 in the Client Hello. The TOE always presents the Signature Algorithms Extension indicating support for P-384."**

Verdict:

PASS.

4.1.3.12.15 FCS_TSLC_EXT.1.6 AGD

If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall verify that AGD guidance includes configuration of the list of supported ciphersuites.

Evaluator Findings:

If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall verify that AGD guidance includes configuration of the list of supported ciphersuites.

The relevant information is found in the following section(s): **"Audit Logging"**

Upon investigation, the evaluator found that the AGD activity states that: **To modify the logging parameters of the TOE, log into the GSS GUI and navigate to the following:**

'System Settings > Logging'

and scroll down to each of the following settings:

- **'Log shipping TLS cipher' – Default is TLS 1.2: AES128-GCM-SHA256**

Must be configured to one of:

TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384

TLS 1.3: TLS_AES_256_GCM_SHA384

Verdict:

PASS.

4.1.3.12.16 FCS_TLSC_EXT.1.7 AGD

None.

4.1.3.12.17 FCS_TLSC_EXT.1.8 AGD

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

Evaluator Findings:
N/A – **The TOE does not support PSKs for TLS.**

Verdict:

PASS.

4.1.3.12.18 FCS_TLSC_EXT.1.9 AGD

None.

4.1.3.13 FCS_TLSS_EXT.1 TLS SERVER PROTOCOL (CPP_ND_V3.0E)

4.1.3.13.1 FCS_TLSS_EXT.1.1 TSS

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified.

Evaluator Findings:
The evaluator checked the description of the implementation of this protocol in the TSS and ensured that the ciphersuites supported are specified. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **The TOE acts as a TLS/HTTPS server to provide a web GUI to administrators. This server supports TLSv1.2 and TLSv1.3 with the following ciphersuites:** • **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384** • **TLS_AES_256_GCM_SHA384** **These ciphersuites are preconfigured in the TOE and can't be configured by administrators.**

The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component.

Evaluator Findings:
The evaluator checked the TSS and ensured that the ciphersuites specified are identical to those listed for this component. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **This server supports TLSv1.2 and TLSv1.3 with the following ciphersuites:** • **TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384** • **TLS_AES_256_GCM_SHA384** **These are identical to the claims made in the SFR and are appropriate for the TOE components.**

The evaluator shall verify that the TSS contains a description of how the TOE technically prevents the use of unsupported and undefined SSL and TLS versions.

Evaluator Findings:
The evaluator verified that the TSS contains a description of how the TOE technically prevents the use of unsupported and undefined SSL and TLS versions. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"** Upon investigation, the evaluator found that the TSS states that**: Any SSL/TLS version stated in the Client Hello other than TLS v1.2 or TLS v1.3 causes the connection to be terminated.**

Verdict:

PASS.

4.1.3.13.2 FCS_TLSS_EXT.1.2 TSS

The evaluator shall verify that the TSS describes the algorithms and key sizes the TSF supports for authenticating itself to TLS clients.

Evaluator Findings:
The evaluator verified that the TSS describes the algorithms and key sizes the TSF supports for authenticating itself to TLS clients. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **The TOE authenticates itself with X.509 certificates using ECDSA over NIST curves secp384r1 or secp521r1.**

The evaluator shall ensure these algorithms are consistent with the selected ciphersuites.

Evaluator Findings:
The evaluator ensured these algorithms are consistent with the selected ciphersuites. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"**

Upon investigation, the evaluator found that the TSS states that: **The TOE authenticates itself with X.509 certificates using ECDSA over NIST curves secp384r1 or secp521r1. These are consistent with the claims made as selections in the SFR.**

Verdict:

PASS.

4.1.3.13.3 FCS_TLSS_EXT.1.3 TSS

The evaluator shall verify that the TSS lists all EC Diffie-Hellman curves and/or Diffie-Hellman groups used in the key establishment by the TOE when acting as a TLS Server. The evaluator shall ensure these algorithms are consistent with the selected ciphersuites.

Evaluator Findings:
The evaluator verified that the TSS lists all EC Diffie-Hellman curves and/or Diffie-Hellman groups used in the key establishment by the TOE when acting as a TLS Server. The evaluator shall ensure these algorithms are consistent with the selected ciphersuites. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Key Establishment is performed using Elliptic Curve Diffie-Hellman P-384 and P-521 keys.** **These are consistent with claims made in the SFR.**

Verdict:

PASS.

4.1.3.13.4 FCS_TLSS_EXT.1.4 TSS

The evaluator shall verify that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246), if session resumption based on session tickets is supported (RFC 5077) and/or if session resumption according to RFC 8446 is supported.

Evaluator Findings:
The evaluator shall verified that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246), if session resumption based on session tickets is supported (RFC 5077) and/or if session resumption according to RFC 8446 is supported. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Session resumption is not supported.**

If session tickets are supported, the evaluator shall verify that the TSS describes that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption.

Evaluator Findings:
If session tickets are supported, the evaluator verified that the TSS describes that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Session tickets are not supported.**

The evaluator shall verify that the TSS identifies the key lengths and algorithms used to protect session tickets.

Evaluator Findings:
The evaluator verified that the TSS identifies the key lengths and algorithms used to protect session tickets. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Session tickets are not supported.**

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

Evaluator Findings:
If session tickets are supported, the evaluator verified that the TSS describes that session tickets adhere to the structural format provided in Section 4 of RFC 5077 and if not, a justification was given of the actual session ticket format. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Session tickets are not supported.**

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

Evaluator Findings:
If the TOE claims a TLS server capable of session resumption (as a single context, or across multiple contexts), the evaluator verified that the TSS describes how session resumption operates (i.e. what would trigger a full handshake, e.g. checking session status, checking Session ID, etc.). If multiple contexts are used, the TSS describes how session resumption is coordinated across those contexts. In case session establishment and session resumption are always using a separate context, the TSS describes how the contexts interact with respect to session resumption (in particular regarding the session ID). It is acceptable for sessions established in one context to be resumable in another context. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Session resumption is not supported.**

Verdict:

PASS.

4.1.3.13.5 FCS_TLSS_EXT.1.5 TSS

The evaluator shall verify that TSS describes whether the list of supported ciphersuites can be configured or not.

Evaluator Findings:
The evaluator shall verify that TSS describes whether the list of supported ciphersuites can be configured or not. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **These ciphersuites are preconfigured in the TOE and can't be configured by administrators.**

Verdict:

PASS.

4.1.3.13.6 FCS_TLSS_EXT.1.6 TSS

None.

4.1.3.13.7 FCS_TLSS_EXT.1.7 TSS

The evaluator shall verify in the TSS that, for TLS 1.3, the TOE shall not permit out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration.

Evaluator Findings:
The evaluator verified in the TSS that, for TLS 1.3, the TOE shall not permit out-of-band provisioning of pre-shared keys (PSKs) in the evaluated configuration. The relevant information is found in the following section(s): **TOE Summary Specification section 6.3.10 labeled "FCS_TLSS_EXT.1, FCS_HTTPS_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **PSK functionality is not supported.**

Verdict:

PASS.

4.1.3.13.8 FCS_TLSS_EXT.1.8 TSS

None.

4.1.3.13.9 FCS_TLSS_EXT.1.1 AGD

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

Evaluator Findings:
The evaluator checked the guidance documentation and ensured that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements). The relevant information is found in the following section(s): **"Cryptographic Functions"** Upon investigation, the evaluator found that the AGD activity states that: **When the TOE acts as a TLS/HTTPS server to provide the administrative web GUI, TLSv1.2 and TLSv1.3 are supported and use the following ciphersuites:** **TLS 1.2: ECDHE_ECDSA_AES256_GCM_SHA384** **TLS 1.3: TLS_AES_256_GCM_SHA384**

Verdict:

PASS.

4.1.3.13.10 FCS_TLSS_EXT.1.2 AGD

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

Evaluator Findings:
The evaluator verified that any configuration necessary to meet the requirement must be contained in the AGD guidance. The relevant information is found in the following section(s): **"Cryptographic Functions"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE performs cryptographic functionality in support of authentication and logging actions. The following key establishment parameters are implemented by the TOE:** **a) TLS Client to syslog server – Elliptic-curve (P-384)** **b) TLS Server for administrative GUI - Elliptic-curve (P-384, P-521)**

Verdict:

PASS.

4.1.3.13.11 FCS_TLSS_EXT.1.3 AGD

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

Evaluator Findings:
The evaluator verified that any configuration necessary to meet the requirement must be contained in the AGD guidance. The relevant information is found in the following section(s): **"Cryptographic Functions"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE performs cryptographic functionality in support of authentication and logging actions. The following key establishment parameters are implemented by the TOE:** **a) TLS Client to syslog server – Elliptic-curve (P-384)** **b) TLS Server for administrative GUI - Elliptic-curve (P-384, P-521)**

Verdict:

PASS.

4.1.3.13.12 FCS_TLSS_EXT.1.4 AGD

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

Evaluator Findings:
The evaluator verified that any configuration necessary to meet the requirement must be contained in the AGD guidance. The relevant information is found in the following section(s): **"Administration Interfaces"** Upon investigation, the evaluator found that the AGD activity states that: **Only the following administration interfaces may be used:** **a) GoSilent Server GUI. Administrative GUI interface for managing all GoSilent configurations on the server and other GoSilent platforms.** **o Local access is provided for the configured whitelisted source IP address of an administrator device, and by navigating to the local IP address of GSS through the (virtual) MGMT Ethernet interface using HTTPS via web browser to TCP port 4439.** **o Remote access is provided to administrators connecting from Cube clients (for any client that has a valid GSS administrator account) using HTTPS via web browser to the MGMT IP address of the GSS over TCP port 443.** **b) GoSilent Cube GUI. User GUI interface that provides limited functionality for initial Cube configuration.** **o Local access is provided to users by navigating to the local IP address of the cube using HTTPS via web browser over port 4439 or by navigating to the URL https://setup.gosilent.** **o Configuration of some Cube functions is done via the GSS GUI. The Cube subsequently downloads these configuration settings from GSS upon connection establishment.**

Verdict:

PASS.

4.1.3.13.13 FCS_TLSS_EXT.1.5 AGD

If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall verify that AGD guidance includes configuration of the list of supported ciphersuites.

Evaluator Findings:
The TOE does not provide the ability to configure the list of supported ciphersuites. No additional configuration instruction is needed.

Verdict:

PASS.

4.1.3.13.14 FCS_TLSS_EXT.1.6 AGD

None.

4.1.3.13.15 FCS_TLSS_EXT.1.7 AGD

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

Evaluator Findings:
The TOE does not use PSKs for TLS. No additional configuration is required.

Verdict:

PASS.

4.1.3.13.16 FCS_TLSS_EXT.1.8 AGD

None.

4.1.4 USER DATA PROTECTION (FDP)

4.1.4.1 FDP_RIP.2 TSS

"Resources" in the context of this requirement are network packets being sent through (as opposed to "to", as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet. The evaluator shall check to ensure that the TSS describes packet processing to the extent that they can determine that no data will be reused when processing network packets. The evaluator shall ensure that this description at a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.

Evaluator Findings:
The evaluator reviewed the TSS, **section labeled "User Data Protection (FDP)"** and verified that it provides a description of packet processing to the extent that it can be determined that no data will be reused when processing network packets. The evaluator also ensured that this description at a minimum describes how the previous data is zeroized/overwritten and at what point in the buffer process this occurs. The evaluator found that the TSS states**: The only resource made available to information flowing through a TOE is the temporary storage of packet information when access is requested and when information is being**

routed. User data is not persistent when resources are released by one user/process and allocated to another user/process. Temporary storage (memory) used to build network packets is erased when the resource is released. Therefore, no residual information from packets in a previous information stream can traverse through the TOE.

4.1.5 STATEFUL TRAFFIC FILTERING (MOD_CPP_FW_V1.4E)

4.1.5.1 FIREWALL (FFW)

FFW_RUL_EXT.1 Stateful Traffic Filtering (MOD_CPP_FW_V1.4E)

4.1.5.1.1 FFW_RUL_EXT.1 TSS

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

Evaluator Findings:

The evaluator reviewed the TSS, **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"** and verified that it provides a description of the TOE's initialization/startup process, which clearly indicates where processing of network packets begins to take place, and provides a discussion that supports the assertion that packets cannot flow during this process.

The evaluator found that the TSS states**: The TOE performs stateful packet filtering on all packets received from or being sent to the External Network, MGMT interface or WAN interface (excepting IPsec traffic) of the GoSilent Server.**

The boot sequence of the TOE aids in establishing the secure domain and preventing tampering or bypass of security functionality. This includes ensuring the packet filtering rules cannot be bypassed during the boot sequence of the TOE. The following steps list the boot sequence for the TOE:

- **BIOS hardware and memory checks**
- **Loading and initialization of the OS**
- **Self-tests including firmware integrity tests are executed**
- **The init utility is started (mounts file systems, sets up network cards, and generally starts all the processes that usually are run on the system at startup)**
- **Daemon programs such as Internet Service Daemon (INETD), Syslogd are started; Routing and forwarding tables are initialized**
- **Application daemons are loaded, enabling access to the GoSilent Server management GUI**
- **Physical interfaces are active**

Once the interfaces are brought up, they will start to receive and send packets based on the current configuration (or not receive or send any packets if they have not been previously configured). Interfaces are brought up only after successful loading of the kernel and daemons, and these interfaces cannot send or receive packets unless previously configured by an administrator. Since the daemon for the management GUI is not loaded until after the kernel and INETD are initialized, no modification to the security attributes can be made by a user or process other than via the management process.

The TOE applies a uniform policy to the traffic flows to and from all GoSilent Cube users. By default, no traffic is allowed to flow from GoSilent Cube users to the External Network, no traffic is allowed to flow from the External Network to GoSilent Cube users, and no traffic is allowed to flow between GoSilent Server interfaces.

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

Evaluator Findings:

The evaluator reviewed the TSS **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"** and verified that it includes a narrative that identifies the components (e.g., active entity such as a process or task) involved in processing the network packets and describes the safeguards that would prevent packets from flowing through the TOE without applying the ruleset in the event of a component failure.

The evaluator found that the TSS states: **The TOE performs stateful packet filtering on all packets received from or being sent to the External Network, MGMT interface or WAN interface (excepting IPsec traffic) of the GoSilent Server.**

The boot sequence of the TOE aids in establishing the secure domain and preventing tampering or bypass of security functionality. This includes ensuring the packet filtering rules cannot be bypassed during the boot sequence of the TOE. The following steps list the boot sequence for the TOE:

- **BIOS hardware and memory checks**
- **Loading and initialization of the OS**
- **Self-tests including firmware integrity tests are executed**
- **The init utility is started (mounts file systems, sets up network cards, and generally starts all the processes that usually are run on the system at startup)**
- **Daemon programs such as Internet Service Daemon (INETD), Syslogd are started; Routing and forwarding tables are initialized**
- **Application daemons are loaded, enabling access to the GoSilent Server management GUI**
- **Physical interfaces are active**

Once the interfaces are brought up, they will start to receive and send packets based on the current configuration (or not receive or send any packets if they have not been previously configured). Interfaces are

brought up only after successful loading of the kernel and daemons, and these interfaces cannot send or receive packets unless previously configured by an administrator. Since the daemon for the management GUI is not loaded until after the kernel and INETD are initialized, no modification to the security attributes can be made by a user or process other than via the management process.

The TOE applies a uniform policy to the traffic flows to and from all GoSilent Cube users. By default, no traffic is allowed to flow from GoSilent Cube users to the External Network, no traffic is allowed to flow from the External Network to GoSilent Cube users, and no traffic is allowed to flow between GoSilent Server interfaces.

The description shall also include a description how the TOE behaves in the situation where the traffic exceeds the amount of traffic the TOE can handle and how it is ensured that also in this condition stateful traffic filtering rules are still applied so that traffic does not pass that shouldn't pass according to the specified rules.

Evaluator Findings:

The evaluator reviewed the TSS **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"** and verified that it includes a description of how the TOE behaves in the situation where the traffic exceeds the amount of traffic the TOE can handle and how it is ensured that also in this condition stateful traffic filtering rules are still applied so that traffic does not pass that shouldn't pass according to the specified rules.

The evaluator found that the TSS states: **In situations where the TOE receives packets faster than they can be processed (flooding), the TOE silently discards the excess packets. An audit record is generated when flooding is detected, but not for each packet.**

Verdict:

PASS.

4.1.5.1.2 FFW_RUL_EXT.1 AGD

The guidance documentation associated with this requirement is assessed in the subsequent test evaluation activities.

Evaluator Findings:

The evaluator reviewed the guidance documentation associated with this requirement and ensured that it is assessed in the subsequent test evaluation activities.

Verdict:

PASS.

4.1.5.1.3 FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4 TSS [TD0924]

The evaluator shall verify that the TSS describes a stateful packet filtering policy and the following attributes are identified as being configurable within stateful traffic filtering rules for the associated protocols:

- ICMPv4
    - Type
    - Code
- ICMPv6
    - Type
    - Code
- IPv4
    - Source address
    - Destination Address
    - Transport Layer Protocol
- IPv6
    - Source Address
    - Destination Address
    - Transport Layer Protocol and where defined by the ST author, Extension Header Type, Extension Header Fields
- TCP
    - Source Port
    - Destination Port
- UDP
    - Source Port
    - Destination Port

Evaluator Findings:
The evaluator reviewed the TSS, **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL.1"**, and ensured that it describes a stateful packet filtering policy and the following attributes are identified as being configurable within stateful traffic filtering rules for the associated protocols: The evaluator found that the TSS states: **By default, the TOE allows administrators to define traffic filtering rules based on a distinct interface and the following network protocol fields:** • **ICMPv4 (Type, Code)** • **ICMPv6 (Type, Code)** • **IPv4 (Source address, Destination address, Transport Layer Protocol)** • **IPv6 (Source address, Destination address, Transport Layer Protocol)** • **TCP (Source port, Destination port)** • **UDP (Source port, Destination port)**

If the TOE does not support the configuration of all possible values in the specified network protocol fields, the evaluator shall verify the TSS identifies which fields cannot filter all values, and for those fields which values cannot be filtered (implicitly or explicitly) by rules.

Evaluator Findings:
The TOE supports the configuration of all possible values in the specified network protocol fields; hence, this assurance activity is not applicable.

For any field and value combinations that are not supported, the evaluator shall confirm the values are not IANA recognized values.

Evaluator Findings:
The TOE supports the configuration of all possible fields and value combinations; hence, this assurance activity is not applicable.

If the TOE supports the implicit configuration of any values, the evaluator shall ensure the TSS identifies the specific configurations that result in the implicit configuration of a value (e.g., configuring ICMP type 8 implicitly configures ICMP code 0).

Evaluator Findings:
The TOE does not support the implicit configuration of any values; hence, this assurance activity is not applicable.

The evaluator shall verify that each rule can identify the following actions: permit or drop with the option to log the operation.

Evaluator Findings:
The evaluator reviewed the TSS **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL.1"** and ensured that each rule can identify the following actions: permit or drop with the option to log the operation. **The TOE allows permit, deny, and log operations to be associated with rules and these rules can be assigned to distinct network interfaces.**

The evaluator shall verify that the TSS identifies all interface types subject to the stateful packet filtering policy and explains how rules are associated with distinct network interfaces.

Evaluator Findings:
The evaluator reviewed the TSS **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL.1"** and ensured that it identifies all interface types subject to the stateful packet filtering policy and explains how rules are associated with distinct network interfaces. **By default, the TOE allows administrators to define traffic filtering rules based on a distinct interface and the following network protocol fields:** • **ICMPv4 (Type, Code)** • **ICMPv6 (Type, Code)** • **IPv4 (Source address, Destination address, Transport Layer Protocol)** • **IPv6 (Source address, Destination address, Transport Layer Protocol)**

- **TCP (Source port, Destination port)**
- **UDP (Source port, Destination port)**

The TOE maintains a session table which tracks all known TCP and UDP sessions based on the information in incoming packets. Specifically, the lookup is based on an exact match of the following network packet attributes:

- **TCP**
- **Source and Destination IP address**
- **Source and Destination Port**
- **Protocol Flags (e.g. SYN, ACK, RST, and FIN flags),**
- **Sequence numbers**
    - **UDP**
- **Source and Destination IP address**
- **Source and Destination Port**

The TSF allows the definition of packet filtering rules by using the following network protocols and protocol fields:

- **IPv4 (RFC 791)**
- **Source address**
- **Destination address**
- **Protocol**
    - **IPv6 (RFC 8200)**
- **Source address**
- **Destination address**
- **Next header (protocol)**
    - **TCP (RFC 793)**
- **Source port**
- **Destination port**
    - **UDP (RFC 768)**

- **Source port**
    - **Source address**
    - **Destination address**
    - **Protocol**
        - **IPv6 (RFC 8200)**
    - **Source address**
    - **Destination address**
    - **Next header (protocol)**
        - **TCP (RFC 793)**
    - **Source port**
    - **Destination port**
        - **UDP (RFC 768)**
    - **Source port**

Verdict:

PASS.

4.1.5.1.4 FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4 AGD [TD0924]

The evaluators shall verify that the guidance documentation identifies the following attributes as being configurable within stateful traffic filtering rules for the associated protocols:

- ICMPv4
    - Type
    - Code
- ICMPv6
    - Type
    - Code
- IPv4

- Source address
- Destination Address
- Transport Layer Protocol

- IPv6
  - Source Address
  - Destination Address
  - Transport Layer Protocol and where defined by the ST author, Extension Header Type, Extension Header Fields

- TCP
  - Source Port
  - Destination Port

- UDP
  - Source Port
  - Destination Port

Evaluator Findings:
The evaluator reviewed the guidance documentation section **"Firewall Configuration"** and confirmed that it identifies the following attributes as being configurable within stateful traffic filtering rules for the associated protocols: **The TOE performs stateful packet filtering via iptables on all packets sent and received from the External Network, MGMT, and WAN interfaces and applies a uniform policy on all traffic to and from GoSilent Cube users.** **To modify the Firewall configurations, navigate to:** **'Firewall > Profile and Options'** **Ensure the following settings are configured:** • **'Active Firewall Profile' – Custom Mode must be selected.** • **'Inbound Firewall Option' – Must be enabled prior to selecting Custom Mode.** **Administrators can define stateful traffic filtering rules based on a distinct interface in addition to the following network protocol fields:** • **ICMPv4 (Type, Code)** • **ICMPv6 (Type, Code)**

- **IPv4 (Source address, Destination address, Transport Layer Protocol)**
- **IPv6 (Source address, Destination address, Transport Layer Protocol)**
- **TCP (Source port, Destination port)**
- **UDP (Source port, Destination port)**

If the TOE does not support the configuration of rules to filter all possible values in the fields identified in FFW_RUL_EXT.1.2, the evaluator shall verify the Guidance describes which field and value combinations are not filterable.

Evaluator Findings:
The evaluator reviewed the guidance documentation and confirmed that there are no values in the fields identified in FFW_RUL_EXT.1.2, hence, this assurance activity is not applicable.

If the TOE supports the implicit configuration of any values, the evaluator shall ensure the Guidance identifies the specific configurations that result in the implicit configuration of a value (e.g., configuring ICMP type 8 implicitly configures ICMP code 0).

Evaluator Findings:
The TOE does not support the implicit configuration of any values; hence, this assurance activity is not applicable.

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

Evaluator Findings:
The evaluator reviewed the guidance documentation and confirmed that it indicates that each rule can identify the following actions: **permit, drop, and log. The TOE Custom chains can also be created to save rules in. Each chain in the Filter table has a 'policy', which is the default action taken on packets processed by the chain. The following policies are available:** **a) DROP. Drops a packet without informing the client.** **b) REJECT. Drops a packet and informs the sender.**

c)	**ACCEPT. Allows the packets to pass the firewall.**

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

Evaluator Findings:
The evaluator reviewed the guidance documentation and confirmed that it explains how rules are associated with distinct network interfaces. **The TOE performs stateful packet filtering via iptables on all packets sent and received from the External Network, MGMT, and WAN interfaces and applies a uniform policy on all traffic to and from GoSilent Cube users.**

Verdict:

PASS.

4.1.5.1.5 FFW_RUL_EXT.1.5 TSS

The evaluator shall verify that the TSS identifies the protocols that support stateful session handling.

Evaluator Findings:
The evaluator reviewed the TSS, **section 6.5.1. FFW_RUL_EXT.1 & FPF_RUL_EXT.1** and verified that it identifies the protocols that support stateful session handling. **The TOE performs stateful packet filtering on all packets received from or being sent to the External Network, MGMT interface or WAN interface (excepting IPsec traffic) of the GoSilent Server.** **By default, the TOE allows administrators to define traffic filtering rules based on a distinct interface and the following network protocol fields:** • **ICMPv4 (Type, Code)** • **ICMPv6 (Type, Code)** • **IPv4 (Source address, Destination address, Transport Layer Protocol)** • **IPv6 (Source address, Destination address, Transport Layer Protocol)** • **TCP (Source port, Destination port)** • **UDP (Source port, Destination port)**

The TSS shall identify TCP, UDP, and, if selected by the ST author, also ICMP.

Evaluator Findings:
The evaluator reviewed the TSS it states: **The TOE maintains a session table which tracks all known TCP and UDP sessions based on the information in incoming packets. Specifically, the lookup is based on an exact match of the following network packet attributes:** • **TCP** • **Source and Destination IP address** • **Source and Destination Port** • **Protocol Flags (e.g. SYN, ACK, RST, and FIN flags),** • **Sequence numbers**   • **UDP** • **Source and Destination IP address** • **Source and Destination Port**

The evaluator shall verify that the TSS describes how stateful sessions are established (including handshake processing) and maintained.

Evaluator Findings:
The evaluator reviewed the TSS and verified that it describes how stateful sessions are established (including handshake processing) and maintained. **The TOE performs stateful network traffic filtering on network packets using the network traffic protocols and network fields specified in FFW_RUL_EXT.1.5 and FPF_RUL_EXT.1.3.**

The evaluator shall verify that for TCP, the TSS identifies and describes the use of the following attributes in session determination: source and destination addresses, source and destination ports, sequence number, and individual flags.

Evaluator Findings:

The evaluator reviewed the TSS and verified that for TCP, it identifies and describes the use of the following attributes in session determination: source and destination addresses, source and destination ports, sequence number, and individual flags.

The evaluator found that the TSS states: **The TOE maintains a session table which tracks all known TCP and UDP sessions based on the information in incoming packets. Specifically, the lookup is based on an exact match of the following network packet attributes:**

- **TCP**
  - **Source and Destination IP address**
  - **Source and Destination Port**
  - **Protocol Flags (e.g. SYN, ACK, RST, and FIN flags),**
  - **Sequence numbers**
- **UDP**
  - **Source and Destination IP address**
  - **Source and Destination Port**

The TSF allows the definition of packet filtering rules by using the following network protocols and protocol fields:

- **IPv4 (RFC 791)**
  - **Source address**
  - **Destination address**
  - **Protocol**
- **IPv6 (RFC 8200)**
  - **Source address**
  - **Destination address**
  - **Next header (protocol)**
- **TCP (RFC 793)**
  - **Source port**
  - **Destination port**
- **UDP (RFC 768)**
  - **Source port**
  - **Destination port**

Conformance with the above listed RFC's has been determined through testing and development and as a requirement of this evaluation.

The evaluator shall verify that for UDP, the TSS identifies and describes the following attributes in session determination: source and destination addresses, source and destination ports.

Evaluator Findings:
The evaluator reviewed the TSS and verified that for UDP, it identifies and describes the following attributes in session determination: **The TOE maintains a session table which tracks all known TCP and UDP sessions based on the information in incoming packets. Specifically, the lookup is based on an exact match of the following network packet attributes:** ▪ **TCP** ○ **Source and Destination IP address** ○ **Source and Destination Port** ○ **Protocol Flags (e.g. SYN, ACK, RST, and FIN flags),** ○ **Sequence numbers** ▪ **UDP** ○ **Source and Destination IP address** ○ **Source and Destination Port**

The evaluator shall verify that for ICMP (if selected), the TSS identifies and describes the following attributes in session determination: source and destination addresses, other attributes chosen in FFW_RUL_EXT.1.5.

Evaluator Findings:
ICMP is not supported by the TOE (only ICMP filtration is present); hence, this activity is not applicable.

The evaluator shall verify that the TSS describes how established stateful sessions are removed.

Evaluator Findings:
The evaluator reviewed the TSS and verified that it describes how established stateful sessions are removed. **For TCP sessions, the TOE removes existing traffic flows from the list of established sessions based on the completion of the session. An exchange of FIN flags indicates the end of data transmission to finish a TCP connection, at which point the session is terminated and the session is removed from the table. TCP session are also removed after a configurable session inactivity timeout threshold. By default, this threshold is**

432000 seconds. Once this timeout period has been reached, the TOE will then terminate the connection and remove it from the table. New packets that match the session will start a new session.

UDP sessions are removed after the session inactivity timeout threshold. Once this threshold has been reached, the TOE terminates the connection and removes it from the table. New packets that match the session will start a new session.

The TSS shall describe how connections are removed for each protocol based on normal completion and/or timeout conditions.

Evaluator Findings:

The evaluator reviewed the TSS and verified that it describes how connections are removed for each protocol based on normal completion and/or timeout conditions. Connections are removed based on:

The TOE tracks the number of half-open TCP connections. When the configured limit for half-open TCP connections is exceeded, TCP SYN are discarded until the number drops below the configured limit. An audit record is generated if the firewall rule is configured to log these events. Half-open TCP connections are automatically discarded after a 60 second timeout period. The following default timeout parameters (in seconds) are configurable in GoSilent Server within the Networking tab:

- **Generic IP Timeout: 600**
- **TCP Connection Timeout: 432000**
- **UDP Protocol Timeout: 30**
- **UDP Stream Timeout: 180**

The TSS shall also indicate when session removal becomes effective (e.g., before the next packet that might match the session is processed).

Evaluator Findings:

The evaluator reviewed the TSS and verified that it indicates when session removal becomes effective (e.g., before the next packet that might match the session is processed).

- **TCP Sessions:**

- **Completion of the session: An exchange of FIN flags indicates the end of data transmission to finish a TCP connection, at which point the session is terminated and removed from the session table.**
- **Session Inactivity Timeout: TCP sessions are removed after a configurable session inactivity timeout threshold. By default, this threshold is 432000 seconds (5 days). Once this timeout period is reached, the TOE terminates the connection and removes it from the table.**

- **UDP Sessions:**
  - **Session Inactivity Timeout: UDP sessions are removed after the session inactivity timeout threshold. Once this threshold has been reached, the TOE terminates the connection and removes it from the table.**

Verdict:

PASS.

4.1.5.1.6 FFW_RUL_EXT.1.5 AGD

The evaluator shall verify that the guidance documentation describes stateful session behaviours. For example, a TOE might not log packets that are permitted as part of an existing session.

Evaluator Findings:
The evaluator reviewed the guidance documentation **"Firewall Configuration"** and verified that it describes stateful session behaviours. **All traffic that correlates to an existing session is permitted, and any traffic not related to a known session is subject to processing by rules that apply flow policies in an administrator-defined order. Rules can be inserted into the ruleset via GUI. Rules are associated to specific interfaces by defining the interface name in the rule. The GUI displays the order of the rules. If no rule matching the traffic exists, the traffic is denied. By default, all traffic from GoSilent Cube users to the External Network, from External Network to Cube users, and between GSS interfaces is denied. A default rule should be applied in the filter table that denies IPv4 and IPv6 source addresses that match the configured IP address on the TOE's interfaces.** **The TOE tracks the number of half-open sessions to an administrator-defined threshold. Once this threshold is met, the TOE will discard TCP SYN packets and log the event until the number is below the threshold. By default, half-open TCP connections are discarded after 60 seconds.**

Verdict:

PASS.

4.1.5.1.7 FFW_RUL_EXT.1.6 TSS

The evaluator shall verify that the TSS identifies the following as packets that will be automatically dropped and are counted or logged:

a) Packets which are invalid fragments, including a description of what constitutes an invalid fragment

b) Fragments that cannot be completely re-assembled

c) Packets where the source address is defined as being on a broadcast network

d) Packets where the source address is defined as being on a multicast network

e) Packets where the source address is defined as being a loopback address

f) The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address "reserved for future use" (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;

g) The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;

h) Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified

i) Other packets defined in FFW_RUL_EXT.1.6 (if any)

Evaluator Findings:
The evaluator reviewed the TSS, **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"** and verified that it identifies the following as packets that will be automatically dropped and are counted or logged: • **The TOE can enforce the following reject rules with logging on traffic:** • **invalid fragments (counted by TSF);** • **fragmented IP packets which cannot be re-assembled completely (counted by TSF);** • **where the source address is equal to the address of the network interface where the network packet was received;** • **where the source address does not belong to the networks associated with the network interface where the network packet was received;** • **where the source address is defined as being on a broadcast network;** • **where the source address is defined as being on a multicast network;** • **where the source address is defined as being a loopback address;** • **where the source address is a multicast;** • **packets where the source or destination address is a link-local address;** • **where the source or destination address is defined as being an address "reserved for future use" as specified in RFC 5735 for IPv4;**

- **where the source or destination address is defined as an "unspecified address" or an address "reserved for future definition and use" as specified in RFC 3513 for IPv6;**
- **with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified;**
- **where the source address of the network packet is equal to the address of the network interface where the network packet was received;**
- **where the source or destination address of the network packet is a link-local address;**
- **where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received;**
- **when a new TCP connection attempt would exceed the configured limit of half-open TCP connections;**
- **packets are checked for validity. "Invalid fragments" are those that violate these rules:**
- **No overlap**
- **The total fragments in one packet should not be more than 62 pieces**
- **The total length of merged fragments should not larger than 64k**
- **All fragments in one packet should arrive in 2 seconds**
- **The total queued fragments has limitation, depending on the platform**
- **The total number of concurrent fragment processing for different packet has limitations depending on platform**

Verdict:

PASS.

4.1.5.1.8 FFW_RUL_EXT.1.6 AGD

The evaluator shall verify that the guidance documentation describes packets that are discarded and potentially logged by default. If applicable protocols are identified, their descriptions need to be consistent with the TSS.

Evaluator Findings:

The evaluator reviewed the guidance documentation **"Table Chains"** and verified that it describes packets that are discarded and potentially logged by default.

Administrators must configure rules as necessary to ensure the TOE drops and is capable of logging packets in all network traffic that meet the following criteria:

a) Packets where the source address of the network packet is defined as being on a broadcast network (dependent on the TOE's network netmasks, including 255.255.255.255);
b) Packets where the source address of the network packet is defined as being on a multicast network (224.0.0.0/4 for IPv4, ff00::/8 for IPv6);
c) Packets where the source address of the network packet is defined as being a loopback address (127.0.0.1/8 for IPv4, ::1 for IPv6);

d) Packets where the source or destination address of the network packet is defined as being unspecified or an address "reserved for future use" as specified in RFC 5735 for IPv4 (0.0.0.0 or 240.0.0.0/4 for IPv4); **e) Packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" as specified in RFC 3513 for IPv6 (:: for IPv6, or anything that does not include the prefix '001');** **f) Packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;** **g) Packets where the source or destination address of the network packet is a link-local address (169.254.0.0/16 for IPv4, fe80::/64 for IPv6);** **h) Packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received. (dependent on the specific networks configured by the administrator);** **i) Packets where new TCP connections are attempted but exceed the configured limit of half-open TCP connections**

If logging is configurable, the evaluator shall verify that applicable instructions are provided to configure auditing of automatically rejected packets.

Evaluator Findings:
The evaluator reviewed the guidance documentation **"Firewall Configuration"** and verified that applicable instructions are provided to configure auditing of automatically rejected packets. **To modify the Networking parameters of the TOE, log into the GSS GUI and navigate to the following:** **'System Settings > Network'** **and scroll down to each of the following settings:** **o 'Enable firewall input logs' – Must be enabled.** **(Note: Enables logging on the firewall INPUT rules.)** **o 'Enable firewall output logs' – Must be enabled.** **(Note: Enables logging on the firewall OUTPUT rules.)** **o 'Enable firewall forward logs' – Must be enabled.** **(Note: Enables logging on the firewall FORWARD rules.)**

Verdict:

PASS.

4.1.5.1.9 FFW_RUL_EXT.1.7 TSS

The evaluator shall verify that the TSS explains how the following traffic can be dropped and counted or logged:

a) Packets where the source address is equal to the address of the network interface where the network packet was received

b) Packets where the source or destination address of the network packet is a link-local address

c) Packets where the source address does not belong to the networks associated with the network interface where the network packet was received, including a description of how the TOE determines whether a source address belongs to a network associated with a given network interface

Evaluator Findings:
The evaluator reviewed the TSS, **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"**, and verified that it explains how the following traffic can be dropped and counted or logged: **a) Packets where the source address is equal to the address of the network interface where the network packet was received. The TOE can enforce the following reject rules with logging on traffic:** **-where the source address of the network packet is equal to the address of the network interface where the network packet was received;** **b) Packets where the source or destination address of the network packet is a link-local address. The TOE can enforce the following reject rules with logging on traffic:** **- where the source or destination address of the network packet is a link-local address;** **c) Packets where the source address does not belong to the networks associated with the network interface where the network packet was received, including a description of how the TOE determines whether a source address belongs to a network associated with a given network interface. The TOE can enforce the following reject rules with logging on traffic:** **-where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received;** **The TOE logs the actions taken on packets when they are processed by the INPUT, OUTPUT, and FORWARD rulesets.**

Verdict:

PASS.

4.1.5.1.10 FFW_RUL_EXT.1.7 AGD

The evaluator shall verify that the guidance documentation describes how the TOE can be configured to implement the required rules.

Evaluator Findings:
The evaluator reviewed the guidance documentation **"Firewall Configuration"** and verified that it describes how the TOE can be configured to implement the required rules: **Administrators must configure rules as necessary to ensure the TOE drops and is capable of logging packets in all network traffic that meet the following criteria:** **a) Packets where the source address of the network packet is defined as being on a broadcast network (dependent on the TOE's network netmasks, including 255.255.255.255);** **b) Packets where the source address of the network packet is defined as being on a multicast network (224.0.0.0/4 for IPv4, ff00::/8 for IPv6);** **c) Packets where the source address of the network packet is defined as being a loopback address (127.0.0.1/8 for IPv4, ::1 for IPv6);** **d) Packets where the source or destination address of the network packet is defined as being unspecified or an address "reserved for future use" as specified in RFC 5735 for IPv4 (0.0.0.0 or 240.0.0.0/4 for IPv4);** **e) Packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" as specified in RFC 3513 for IPv6 (:: for IPv6, or anything that does not include the prefix '001');** **f) Packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;** **g) Packets where the source or destination address of the network packet is a link-local address (169.254.0.0/16 for IPv4, fe80::/64 for IPv6);** **h) Packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received. (dependent on the specific networks configured by the administrator);** **i) Packets where new TCP connections are attempted but exceed the configured limit of half-open TCP connections**

If logging is configurable, the evaluator shall verify that applicable instructions are provided to configure auditing of automatically rejected packets.

Evaluator Findings:
The evaluator reviewed the guidance documentation and verified that it provides applicable instructions to configure auditing of automatically rejected packets.

An example configuration of this rule could include the following, however it should be noted that the *--limit-burst* and *--limit 1/s* values can adversely affect performance and reachability if inappropriately configured.

-N LOG_THROTTLE_SYN_ACCEPT

-N LOG_THROTTLE_SYN_DROP

-N THROTTLE_SYN

#Throttle SYNs to limit half open TCP connections

-A FORWARD -p tcp --syn -j THROTTLE_SYN

-A THROTTLE_SYN -m limit --limit 1/s --limit-burst 2 -j LOG_THROTTLE_SYN_ACCEPT

 # Log accepted SYNs if necessary. (Uncomment next line to enable logging)

#-A LOG_THROTTLE_SYN_ACCEPT -j LOG --log-prefix "[firewall_custom_log] OK "

-A LOG_THROTTLE_SYN_ACCEPT -j ACCEPT

 # Log rejected SYNs if necessary

-A LOG_THROTTLE_SYN_DROP -j LOG --log-prefix "[firewall_custom_log] DROP "

-A LOG_THROTTLE_SYN_DROP -j DROP

-A THROTTLE_SYN -j LOG_THROTTLE_SYN_DROP

Verdict:

PASS.

4.1.5.1.11 FFW_RUL_EXT.1.8 TSS [TD0545]

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

Evaluator Findings:
The evaluator reviewed the TSS, **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"** and verified that it describes the algorithm applied to incoming packets, including the processing of default rules, determination of whether a packet is part of an established session, and application of an administrator-defined and ordered ruleset.

Once the interfaces are brought up, they will start to receive and send packets based on the current configuration (or not receive or send any packets if they have not been previously configured). Interfaces are brought up only after successful loading of the kernel and daemons, and these interfaces cannot send or receive packets unless previously configured by an administrator. Since the daemon for the management GUI is not loaded until after the kernel and INETD are initialized, no modification to the security attributes can be made by a user or process other than via the management process.

The TOE applies a uniform policy to the traffic flows to and from all GoSilent Cube users. By default, no traffic is allowed to flow from GoSilent Cube users to the External Network, no traffic is allowed to flow from the External Network to GoSilent Cube users, and no traffic is allowed to flow between GoSilent Server interfaces.

By default, the TOE allows administrators to define traffic filtering rules based on a distinct interface and the following network protocol fields:

- **ICMPv4 (Type, Code)**
- **ICMPv6 (Type, Code)**
- **IPv4 (Source address, Destination address, Transport Layer Protocol)**
- **IPv6 (Source address, Destination address, Transport Layer Protocol)**
- **TCP (Source port, Destination port)**
- **UDP (Source port, Destination port)**

The TOE maintains a session table which tracks all known TCP and UDP sessions based on the information in incoming packets. Specifically, the lookup is based on an exact match of the following network packet attributes:

- **TCP**
- **Source and Destination IP address**
- **Source and Destination Port**
- **Protocol Flags (e.g. SYN, ACK, RST, and FIN flags),**
- **Sequence numbers**
- **UDP**
- **Source and Destination IP address**
- **Source and Destination Port**

[TD0545] If the TOE implements a mechanism that ensures that no conflicting rules can be configured, the TSS shall describe the underlying mechanism.

Evaluator Findings:
The TOE does not implement a mechanism to ensure no conflicting rules can be configured; hence, this activity is not applicable.

Verdict:

PASS.

4.1.5.1.12 FFW_RUL_EXT.1.8 AGD

The evaluator shall verify that the guidance documentation describes how the order of stateful traffic filtering rules is determined and provides the necessary instructions so that an administrator can configure the order of rule processing.

Evaluator Findings:
The evaluator reviewed the guidance documentation section **"Firewall Configuration"** and verified that it describes how the order of stateful traffic filtering rules is determined and provides the necessary instructions so that an administrator can configure the order of rule processing. **This section states that rules are processed in the administrator defined order. "Rules can be inserted into the ruleset via GUI. Rules are associated to specific interfaces by defining the interface name in the rule. The GUI displays the order of the rules. If no rule matching the traffic exists, the traffic is denied."**

Verdict:

PASS.

4.1.5.1.13 FFW_RUL_EXT.1.9 TSS

The evaluator shall verify that the TSS describes the process for applying stateful traffic filtering rules and also that the behavior (either by default, or as configured by the administrator) is to deny packets when there is no rule match unless another required conditions allows the network traffic (i.e., FFW_RUL_EXT.1.5 or FFW_RUL_EXT.2.1).

Evaluator Findings:

The evaluator reviewed the TSS, **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"** and verified that it describes the process for applying stateful traffic filtering rules and that the behavior (either by default or as configured by the administrator) is to deny packets when there is no rule match unless another required condition allows the network traffic (i.e., FFW_RUL_EXT.1.5 or FFW_RUL_EXT.2.1).

Traffic for known sessions is permitted to flow. For traffic not associated with known sessions, rules within information flow policies are processed in an administrator-defined order to determine if the traffic should be forwarded. By default, the TOE behavior is to deny packets when there is no rule match. The TOE performs stateful network traffic filtering on network packets using the network traffic protocols and network fields specified in FFW_RUL_EXT.1.5 and FPF_RUL_EXT.1.3. The TOE allows permit, deny, and log operations to be associated with rules and these rules can be assigned to distinct network interfaces.

Verdict:

PASS.

4.1.5.1.14 FFW_RUL_EXT.1.9 AGD

The evaluator shall verify that the guidance documentation describes the behavior if no rules or special conditions apply to the network traffic.

Evaluator Findings:
The evaluator reviewed the guidance documentation section **"Firewall Configuration"** and verified that it describes the behavior if no rules or special conditions apply to the network traffic. **This section states that if no rule matching the traffic exists, the traffic is denied.**

If the behavior is configurable, the evaluator shall verify that the guidance documentation provides the appropriate instructions to configure the behavior to deny packets with no matching rules.

Evaluator Findings:
The behavior is not configurable; hence, this activity is not applicable.

Verdict:

PASS.

4.1.5.1.15 FFW_RUL_EXT.1.10 TSS

The evaluator shall verify that the TSS describes how the TOE tracks and maintains information relating to the number of half-open TCP connections.

Evaluator Findings:
The evaluator reviewed the TSS, **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"** and verified that it describes how the TOE tracks and maintains information relating to the number of half-open TCP connections. **The TOE tracks the number of half-open TCP connections. When the configured limit for half-open TCP connections is exceeded, TCP SYN are discarded until the number drops below the configured limit. An audit record is generated if the firewall rule is configured to log these events. Half-open TCP connections are automatically discarded after a 60 second timeout period.**

The TSS should identify how the TOE behaves when the administratively defined limit is reached and should describe under what circumstances stale half-open connections are removed (e.g. after a timer expires).

Evaluator Findings:
The evaluator verified that the TSS identifies how the TOE behaves when the administratively defined limit is reached and describes under what circumstances stale half-open connections are removed (e.g., after a timer expires). **When the configured limit for half-open TCP connections is exceeded, TCP SYN are discarded until the number drops below the configured limit. An audit record is generated if the firewall rule is configured to log these events.**

Verdict:

PASS.

4.1.5.1.16 FFW_RUL_EXT.1.10 AGD

The evaluator shall verify that the guidance documentation describes the behaviour of imposing TCP half-open connection limits and its default state if unconfigured. The evaluator shall verify that the guidance clearly indicates the conditions under which new connections will be dropped e.g. per-destination or per-client.

Evaluator Findings:

The evaluator reviewed the guidance documentation **"Firewall Configuration"** and verified that it describes the behavior of imposing TCP half-open connection limits and its default state if unconfigured. **This section states that the TOE tracks the number of half-open sessions to an administrator defined threshold. When threshold is exceeded, the TOE will discard TCP SYN packets and log the event. The default behaviour of the TOE is to discard half-open TCP connections after 60 seconds.**

Verdict:

PASS.

4.1.6 IDENTIFICATION AND AUTHENTICATION (FIA)

4.1.6.1 FIA_AFL.1 AUTHENTICATION FAILURE MANAGEMENT (CPP_ND_V3.0E)

4.1.6.1.1 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.

Evaluator Findings:
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 relevant information is found in the following section(s): **TOE Summary Specification section 6.6.1 FIA_AFL.1, FIA_PMG_EXT.1, FIA_UIA_EXT.1 & FIA_UAU.7"** Upon investigation, the evaluator found that the TSS states that: **The TOE tracks the number of sequential failed authentication attempts for each user account. Upon meeting the configured limit for failed authentication attempts, the TOE locks the account in question for an administrator configured period of time. During this time, entering the correct password for the locked account will still result in an authentication failure. Any successful authentication resets the counter to zero.**

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).

Evaluator Findings:
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).

The relevant information is found in the following section(s): **TOE Summary Specification FIA_AFL.1, FIA_PMG_EXT.1, FIA_UIA_EXT.1 & FIA_UAU.7.**

Upon investigation, the evaluator found that the TSS states that: **Administrator account lockout is not applicable to local access.**

Verdict:

PASS.

4.1.6.1.2 FIA_AFL.1 AGD

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.

Evaluator Findings:
The evaluator examined 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 relevant information is found in the following section(s): **"Account Lockout"** Upon investigation, the evaluator found that the AGD activity states that: **GSS supports account lockout functions resulting from excessive failed login attempts. To modify these functions and the thresholds of GSS, login to the GSS GUI and navigate to the following:** **'System Settings > Admin Console'** **and scroll down to the following settings for administrator defined configuration:** ○ **'Login attempts before lockout' (Note: Must be set between 1 and 10)** ○ **'Lockout interval' (Note: Must not be 0)** ○ **'Minimum password length' (Note: this parameter should be configured to at least 8 characters)** **The Cube also supports account lockout functions resulting from excessive failed login attempts. To modify these functions and the thresholds of the Cube, log into the GSS GUI and navigate to the following:** **'System Settings > GoSilent'** **and scroll down to the following settings for administrator defined configuration:** ○ **'Login attempts before lockout' (Note: Must be set between 1 and 10)** ○ **'Lockout interval' (Note: Must not be 0)**

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.

Evaluator Findings:
The evaluator examined 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. The relevant information is found in the following section(s): "**Verify Firewall Configuration" and "Restrict No-Lockout Access to Whitelisted IP"** Upon investigation, the evaluator found that the AGD activity: **"Verify Firewall Configuration" which states that administrator accounts are not subject to account locking when using the local access mechanism.**

"Restrict No-Lockout Access" details the instructions on how to configure an admin account to not lose remote access via lockout. To restrict no-lockout access to the configured whitelisted source IP address via the (virtual) MGMT Ethernet interface using HTTPS to TCP port 4439, perform the following steps in the GSS GUI:

a) **Navigate to 'Firewall > Profile and Options' tab within the GSS GUI. Then navigate to 'Inbound Firewall Option' pane and ensure the check box is selected. Click 'Save Inbound Firewall Settings' button to save the selection.**

b) **Next, in the 'Active Firewall Profile' pane, select the 'Custom Mode' radio button. Click 'Save Active Firewall Profile' button to save the selection.**

c) **Navigate to the 'Advanced Rules' tab.**

Note: The necessary rulesets to implement the CC evaluated configuration for local access are already in place by default. This ruleset is labelled in iptables as the 'GSS-MGMT-LOCAL-4439' chain. However, several refinements are required to tailor these rules to the specific network environment.

- **Navigate to rule 45 and substitute the IP address of the allowed administrator system.**
- **Navigate to rule 43 and delete this rule. This rule causes the GSS-MGMT-LOCAL-4430 chain to return before processing the local access configuration rules. Save the configuration.**

Note: Be careful to delete the first instance of this rule in the chain, not the instance at the end of the chain identified as rule 49.

Verdict:

PASS.

4.1.6.2 FIA_PMG_EXT.1 PASSWORD MANAGEMENT (CPP_ND_V3.0E)

4.1.6.2.1 FIA_PMG_EXT.1 TSS

The evaluator shall check that the TSS:

a. lists the supported special character(s) for the composition of administrator passwords.

b. to ensure that the minimum_password_length parameter is configurable by a Security Administrator.

c. lists the range of values supported for the minimum_password_length parameter. The listed range shall include the value of 15.

Evaluator Findings:
The evaluator checked that the TSS: a. lists the supported special character(s) for the composition of administrator passwords. b. to ensure that the minimum_password_length parameter is configurable by a Security Administrator.

c. lists the range of values supported for the minimum_password_length parameter. The listed range shall include the value of 15.

The relevant information is found in the following section(s): **TOE Summary Specification section 6.6.1 FIA_AFL.1, FIA_PMG_EXT.1, FIA_UIA_EXT.1 & FIA_UAU.7"**

Upon investigation, the evaluator found that the TSS states that: **Authentication of an administrator is through use of a username/password. The minimum password may be configured from 8 to 40 characters, that incorporate a combination of lowercase letters, uppercase letters, numbers and special characters ("!", "@", "#", "$", "%", "^", "&", "*", "(", ")"). During entry of the password, each character entered is masked with a "*" when progress is reflected on the screen. If an authentication attempt fails, (either the username is not recognized or the password is incorrect) the same "Login failed" error message is presented.**

Verdict:

PASS.

4.1.6.2.2 FIA_PMG_EXT.1 AGD

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.

Evaluator Findings:
The evaluator examined 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. The relevant information is found in the following section(s): **"Authentication"** Upon investigation, the evaluator found that the AGD activity states that: **To modify the authentication parameters for Cube, log into the GSS GUI and navigate to the following:** **'System Settings > GoSilent'** **and scroll down to the 'Minimum password length' setting and enter a value.** **This parameter should be configured to at least 8 characters, and up to 40 characters.** **The TOE supports the following characters for use in passwords:** a) **Upper and lower case letters** b) **Numbers** c) **Special Characters:** ○ **("!", "@", "#", "$", "%", "^", "&", "*", "(", ")").**

Verdict:

PASS.

4.1.6.3 FIA_PSK_EXT.1 PRE-SHARED KEY COMPOSITION (MOD_VPNGW_V1.3)

4.1.6.3.1 FIA_PSK_EXT.1 TSS (MOD_VPNGW_V1.3)

The evaluator shall confirm that the TSS states which pre-shared key selections are supported for IKEv2 per FCS_IPSEC_EXT.1.13 and FPF_MFA_EXT.1.1.

Evaluator Findings:
The evaluator verified that the TSS, **section 6.6.3 FIA_PSK_EXT.1, FIA_PSK_EXT.2"**, states which pre-shared key selections are supported for IKEv2 per FCS_IPSEC_EXT.1.13 and FPF_MFA_EXT.1.1. **The TOE supports RFC 8784 PPKs with IPsec. Administrators may import (load) externally-generated bit-based PPKs and their associated PPK-IDs onto GoSilent Server and GoSilent Cube, and then configure a PPK-ID to be used with a specific remote endpoint. If an endpoint is configured for a connection, then PPKs must be**

used for that connection and the same PPK-ID must be configured for both GoSilent Server and GoSilent Cube. If no PPK-ID is configured, standard IKEv2 protocols are used for the connection.

Verdict:
PASS.

4.1.6.3.2 FIA_PSK_EXT.1 AGD **[TD0838]** (MOD_VPNGW_V1.3)

The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on how to configure all selected pre-shared key options if any configuration is required.

Evaluator Findings:
The evaluator checked the AGD **"Post-quantum Pre-shared Keys Configuration"** and ensured that it provides guidance to administrators on how to configure all selected pre-shared key options if any configuration is required. **Pre-shared keys are generated on a third-party system.** **PPK_IDs are configured on a per-connection basis. On the GSS, each PPK_ID may be configured for just one VPN Client (e.g. Cube). On the Cube, a single PPK_ID may be configured in more than one profile (e.g. GSS). Since a Cube only has a single active IPsec connection at any time, and any PPK_ID can only be configured for one Cube on each GSS, this scenario ensures that each PPK can only be in use with one IPsec connection at any time.** **Usage of PPKs with GoSilent requires several steps. First, the {PPK_ID, PPK} pairs must be loaded onto the GSS and Cubes. Second, the appropriate PPK_ID must be configured on both the GSS and associated Cube. The third step is optional. On the GSS, a parameter may be set to require all IPsec connections to use PPKs. Any connection attempt from a remote endpoint (e.g. Cube) that does not indicate usage of a known PPK_ID that is configured for that endpoint is rejected. If the two endpoints of a connection are configured to use different PPK_IDs, the connection will fail.** **Under the subsections of "Post-quantum Keys configuration" there is instruction for how the admin can configure these for both of the components: GSS and the Cube.**

The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on how to configure the mandatory_or_not flag per RFC 8784.

Evaluator Findings:
The evaluator checked the AGD and ensured that it provides guidance to administrators on how to configure the mandatory_or_not flag per RFC 8784. The administrator The guidance states that: **Post-quantum Pre-shared Keys (PPKs) as defined in RFC 8784 may be used as an enhancement to the standard ECDHE key establishment process.**

As RFC 8784 relates to the unique ID, configuration can be found for the Server (GSS) under "GSS" and configuration for the Cube is under "Cube". Both of these subsections are listed un "Post-Quantum Pre-shared Keys Configuration."

Further, for making sure the IDs are linked with the IPsec Remote endpoints, further instruction is under "Associating PK IDs with IPsec Remote Endpoints" which is also a subsection of "Post-Quantum Pre-Shared Keys Configuration and includes instructions for both GSS and the Cube.

Due to the length of the instructions the evaluator references these sections and has verified the information to be present.

Verdict:
PASS.

4.1.6.4 FIA_PSK_EXT.2 GENERATED PRE-SHARED KEYS (MOD_VPNGW_V1.3)

4.1.6.4.1 FIA_PSK_EXT.2 TSS (MOD_VPNGW_V.1.3)

If "generate" is selected, the evaluator shall confirm that this process uses the RBG specified in FCS_RBG_EXT.1 and the output matches the size selected in FIA_PSK_EXT.2.1.

Evaluator Findings:
The selection "generate" is not claimed; hence, this activity is not applicable.

Verdict:
PASS.

4.1.6.4.2 FIA_PSK_EXT.2 AGD (MOD_VPNGW_V.1.3)

The evaluator shall confirm the operational guidance contains instructions for entering generated pre-shared keys for each protocol identified in the FIA_PSK_EXT.1.1.

Evaluator Findings:
The evaluator checked the AGD section **"Post-quantum Pre-shared Keys Configuration"** and ensured that it contains instructions for entering generated pre-shared keys for each protocol identified in the FIA_PSK_EXT.1.1. **To load a PPK on the GSS, PPK usage must first be enabled. Log into the GUI and navigate to the following:** **'System Settings'** **Scroll down to the following setting:** **o 'PPK Secret Size' – The default value of "256-bit (Standard)" must always be used.** **o 'Enable PPK support for IKEv2 clients' – Must be enabled.**

This step only needs to be performed once, and the GSS must be restarted after enabling the parameter for it to take effect.

Verdict:
PASS.

4.1.6.5 FIA_UIA_EXT.1 USER IDENTIFICATION AND AUTHENTICATION (CPP_ND_V3.0E)

4.1.6.5.1 FIA_UIA_EXT.1 TSS

The evaluator shall examine the TSS to determine that it describes the logon process for remote authentication mechanism (e.g. SSH public key, Web GUI password, etc.) and optional local authentication mechanisms supported by the TOE. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon".

Evaluator Findings:
The evaluator examined the TSS to determine that it describes the logon process for remote authentication mechanism (e.g. SSH public key, Web GUI password, etc.) and optional local authentication mechanisms supported by the TOE. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon". The relevant information is found in the following section(s): **TOE Summary Specification section 6.6.1 labeled "FIA_AFL.1, FIA_PMG_EXT.1, FIA_UIA_EXT.1 & FIA_UAU.7"** Upon investigation, the evaluator found that the TSS states that: **Identification and authentication are required for both local and remote administrator access. Management access to the TOE is via an HTTPS session.** **Authentication of an administrator is through use of a username/password. If an authentication attempt fails, (either the username is not recognized or the password is incorrect) the same "Login failed" error message is presented.** **The TOE supports local authentication where it looks up the username in its local configuration and compares the hash of the password to the saved value. If the credentials are valid, the user is successfully authenticated and is authorized to access the management interface.**

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

Evaluator Findings:
The evaluator examined the TSS and determined that it describes which actions are allowed before administrator identification and authentication. The description shall cover authentication and identification for local and remote TOE administration. The relevant information is found in the following section(s): **TOE Summary Specification 6.6.1 labeled "FIA_AFL.1, FIA_PMG_EXT.1, FIA_UIA_EXT.1 & FIA_UAU.7"** Upon investigation, the evaluator found that the TSS states that: **For both local and remote connections, the TOE only provides the warning banner as described in FTA_TAB.1 prior to authentication.**

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

Evaluator Findings:
The evaluator examined the TSS and ensured that it details how Security Administrators are authenticated and identified by all TOE components. If not, all TOE components support authentication of Security Administrators according to FIA_UIA_EXT.1, the TSS shall describe how the overall TOE functionality is split between TOE components including how it is ensured that no unauthorized access to any TOE component can occur. The relevant information is found in the following section(s): **TOE Summary Specification 6.6.1 labeled "FIA_AFL.1, FIA_PMG_EXT.1, FIA_UIA_EXT.1 & FIA_UAU.7"** Upon investigation, the evaluator found that the TSS states that: **Management of the TOE is primarily performed locally or remotely through GoSilent Server. However, management of GoSilent Cube locally provides the initial configuration information for it to connect to GoSilent Server.** **Identification and authentication are required for both local and remote administrator access. Management access to the TOE is via an HTTPS session.** **The TOE supports local authentication where it looks up the username in its local configuration and compares the hash of the password to the saved value. If the credentials are valid, the user is successfully authenticated and is authorized to access the management interface.**

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

Evaluator Findings:
The evaluator examined the TSS and ensured that it describes for each TOE component which actions are allowed before administrator identification and authentication. The description shall cover authentication and identification for remote TOE administration and optionally for local TOE administration if claimed by the ST author. For each TOE component that does not support authentication of Security Administrators according to FIA_UIA_EXT.1 the TSS shall describe any unauthenticated services/services that are supported by the component. The relevant information is found in the following section(s): **TOE Summary Specification 6.6.1 "FIA_AFL.1, FIA_PMG_EXT.1, FIA_UIA_EXT.1 & FIA_UAU.7"** Upon investigation, the evaluator found that the TSS states that: **For both local and remote connections, the TOE only provides the warning banner as described in FTA_TAB.1 prior to authentication.**

Verdict:

PASS.

4.1.6.5.2 FIA_UIA_EXT.1 AGD

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.

Evaluator Findings:

The evaluator examined the guidance documentation section **"Administration Interfaces"** and determined 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.
The relevant information is found in the following section(s): **"Administration Interfaces"**
Upon investigation, the evaluator found that the AGD activity states that: **Only the following administration interfaces may be used:**
• **GoSilent Server GUI. Administrative GUI interface for managing all GoSilent configurations on the server and other GoSilent platforms.**
  ○ **Local access is provided for the configured whitelisted source IP address of an administrator device, and by navigating to the local IP address of GSS through the (virtual) MGMT Ethernet interface using HTTPS via web browser to TCP port 4439.**
  ○ **Remote access is provided to administrators connecting from Cube clients (for any client that has a valid GSS administrator account) using HTTPS via web browser to the MGMT IP address of the GSS over TCP port 443.**
• **GoSilent Cube GUI. User GUI interface that provides limited functionality for initial Cube configuration.**
  ○ **Local access is provided to users by navigating to the local IP address of the cube using HTTPS via web browser over port 4439 or by navigating to the URL https://setup.gosilent.**
  ○ **Configuration of some Cube functions is done via the GSS GUI. The Cube subsequently downloads these configuration settings from GSS upon connection establishment.**

Verdict:

PASS.

4.1.6.6 FIA_UAU.7 PROTECTED AUTHENTICATION FEEDBACK (CPP_ND_V3.0E)

4.1.6.6.1 FIA_UAU.7 TSS

None.

4.1.6.6.2 FIA_UAU.7 AGD

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.

Evaluator Findings:
There are no preparatory steps needed to ensure authentication data is not revealed while entering login information locally. Information is obscured by default.

Verdict:

PASS.

4.1.6.7 FIA_X509_EXT.1/REV X.509 CERTIFICATE VALIDATION (CPP_ND_V3.0E)

4.1.6.7.1 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).

Evaluator Findings:

The evaluator ensured 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).

The relevant information is found in the following section(s): **TOE Summary Specification section 6.6.2 labeled "FIA_X509_EXT.1/ITT, FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3"**

Upon investigation, the evaluator found that the TSS states that: **When a TOE component receives a certificate asserting the identity of a remote system during an IPsec connection, the TOE ensures the current time is within the validity time of the certificate, the certificate has not been revoked (verified via OCSP for IPsec or CRLs for TLS), contains the appropriate extendedKeyUsage (EKU) purpose set (Server Authentication), and the certificate chain terminates with a trusted CA certificate. The certificate chain is validated by verifying each certificate (except for the end entity certificate) in the chain is currently valid, has not been revoked, contains the basic constraints extension with the CA flag set to TRUE, and is signed by a trusted CA or is explicitly configured as a trusted CA. The entire chain is validated regardless of whether the only a leaf certificate is presented. If the TOE cannot establish a connection to the OCSP server or CDP, the TOE will reject the certificate. The certificate of the OCSP responder must include OCSP Signing in its EKU.**

Revocation checking for the entire certificate chain (to the Root) is performed when certificates are loaded into the TOE, when certificates are received from the remote side during TLS and IPsec connection establishment, and when OCSP responses are received.

The TOE uses X.509 certificates to:

- **provide IPsec connections between GoSilent Server and Cube devices**
- **verify the identity of the Syslog server**

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.

Evaluator Findings:
The TSS describes 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. The relevant information is found in the following section(s): **TOE Summary Specification section 6.6.2 labeled "FIA_X509_EXT.1/ITT, FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3"** Upon investigation, the evaluator found that the TSS states that: **The certificate chain is validated by verifying each certificate (except for the end entity certificate) in the chain is currently valid, has not been revoked, contains the basic constraints extension with the CA flag set to TRUE, and is signed by a trusted CA or is explicitly configured as a trusted CA. The entire chain is validated regardless of whether the only a leaf certificate is presented. If the TOE cannot establish a connection to the OCSP server, the TOE will reject the certificate. The certificate of the OCSP responder must include OCSP Signing in its EKU. If the TOE cannot establish a connection to the CDP, the TOE will accept the certificate.** **When a TOE component receives a certificate asserting the identity of a remote system during an IPsec connection, the TOE ensures the current time is within the validity time of the certificate, the certificate has not been revoked (verified via OCSP for IPsec or CRLs for TLS), contains the appropriate extendedKeyUsage purpose set (Server Authentication), and the certificate chain terminates with a trusted CA certificate.** **Revocation checking for the entire certificate chain (to the Root) is performed when certificates are loaded into the TOE, when certificates are received from the remote side during TLS and IPsec connection establishment, and when OCSP responses are received.**

Verdict:

PASS.

4.1.6.7.2 FIA_X509_EXT.1/REV AGD

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.

Evaluator Findings:
The evaluator also ensured 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. The relevant information is found in the following section(s): **"Certificate Management"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE leverages X509 certificates to provide IPsec connections between GSS and Cube devices, and to verify the identity of the Syslog server.** **During the enablement process for a Cube, the X509 certificate for GSS and the specific Cube must be communicated out of band and imported by an authorized administrator for each component. This communication should be done via secure communication channel.** **The TOE performs several certificate validity checks on upload, and during TLS and IPsec connection establishment. These checks include a check against its own trust store, the expiration date, revocation status**

via OCSP (IPSec) and CRL (TLS) and verifies the correct extendedKeyUsage purpose set (Server Authentication). During a certificate upload, the TOE also verifies that the certificate chain terminates with a trusted CA certificate. When validating a certificate used for IPsec, the reference identifier will be matched against the DN (See below for configuration). Certificates used for TLS connections will fall back to the CN if the SAN is not present.

Verdict:

PASS.

4.1.6.8 FIA_X509_EXT.1/ITT X.509 CERTIFICATE VALIDATION (CPP_ND_V3.0E)

4.1.6.8.1 FIA_X509_EXT.1/ITT X.509 TSS

The evaluator shall examine the TSS to ensure it describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied).

Evaluator Findings:

The evaluator examined the TSS to ensure it describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied).

The relevant information is found in the following section(s): **TOE Summary Specification section 6.6.2 labeled "FIA_X509_EXT.1/ITT, FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3"**

Upon investigation, the evaluator found that the TSS states that: **When a TOE component receives a certificate asserting the identity of a remote system during an IPsec connection, the TOE ensures the current time is within the validity time of the certificate, the certificate has not been revoked (verified via OCSP for IPsec or CRLs for TLS), contains the appropriate extendedKeyUsage purpose set (Server Authentication), and the certificate chain terminates with a trusted CA certificate. The certificate chain is validated by verifying each certificate (except for the end entity certificate) in the chain is currently valid, has not been revoked, contains the basic constraints extension with the CA flag set to TRUE, and is signed by a trusted CA or is explicitly configured as a trusted CA. The entire chain is validated regardless of whether the only a leaf certificate is presented. If the TOE cannot establish a connection to the OCSP server, the TOE will reject the certificate. The certificate of the OCSP responder must include OCSP Signing in its EKU. If the TOE cannot establish a connection to the CDP, the TOE will accept the certificate.**

Revocation checking for the entire certificate chain (to the Root) is performed when certificates are loaded into the TOE, when certificates are received from the remote side during TLS and IPsec connection establishment, and when OCSP responses are received.

The EKU checks for Code Signing and Client Authentication are not applicable to this TOE. Trusted updates are not verified using X.509 certificates, and TLS mutual authentication is not supported.

If selected, the TSS shall describe how certificate revocation checking is performed. It is expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication).

Evaluator Findings:
The evaluator examined the TSS and ensure that it describes how certificate revocation checking is performed. It is expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). The relevant information is found in the following section(s): **TOE Summary Specification section 6.6.2 labeled "FIA_X509_EXT.1/ITT, FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3"** Upon investigation, the evaluator found that the TSS states that: **Revocation checking for the entire certificate chain (to the Root) is performed when certificates are loaded into the TOE, when certificates are received from the remote side during TLS and IPsec connection establishment, and when OCSP responses are received.** **On GoSilent Server, distinct leaf certificates are supported for IPsec (VPN connections) and the administrator GUI. The same certificate can be used for both functions, or separate certificates can be used and are therefore determined during import when selecting the usage for the certificate. On GoSilent Cube, a distinct leaf certificate is supported for IPsec connections with each configured Server profile. The certificates used is determined by the certificate configured in the profile selected by the user. On the Cube, the client certificate to be used is chosen during the configuration of the server profile (Virtual Server setup). When receiving a certificate, GoSilent Server will verify it against its own trust store, and check the revocation status via OCSP (for IPsec) and CRL (for TLS).**

Verdict:

PASS.

4.1.6.8.2 FIA_X509_EXT.1/ITT X.509 AGD

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

Evaluator Findings:
The evaluator ensured that the guidance documentation describes where the check of validity of the certificates takes place, describes any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) and describe how certificate revocation checking is performed. The relevant information is found in the following section(s): **"Certificate Management"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE leverages X509 certificates to provide IPsec connections between GSS and Cube devices, and to verify the identity of the Syslog server.** **During the enablement process for a Cube, the X509 certificate for GSS and the specific Cube must be communicated out of band and imported by an authorized administrator for each component. This communication should be done via secure communication channel.** **The TOE performs several certificate validity checks on upload, and during TLS and IPsec connection establishment. These checks include a check against its own trust store, the expiration date, revocation status via OCSP (IPSec) and CRL (TLS) and verifies the correct extendedKeyUsage purpose set (Server Authentication). During a certificate upload, the TOE also verifies that the certificate chain terminates with a trusted CA certificate. When validating a certificate used for IPsec, the reference identifier will be matched**

against the DN (See below for configuration). Certificates used for TLS connections will fall back to the CN if the SAN is not present.

Verdict:

PASS.

4.1.6.9 FIA_X509_EXT.2 X.509 CERTIFICATE AUTHENTICATION (CPP_ND_V3.0E)

4.1.6.9.1 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.

Evaluator Findings:
The evaluator checked the TSS and ensured that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates. The relevant information is found in the following section(s): **TOE Summary Specification section 6.6.2 labeled "FIA_X509_EXT.1/ITT, FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3"** Upon investigation, the evaluator found that the TSS states that: **On GoSilent Server, distinct leaf certificates are supported for IPsec (VPN connections) and the administrator GUI. The same certificate can be used for both functions, or separate certificates can be used and are therefore determined during import when selecting the usage for the certificate. On GoSilent Cube, a distinct leaf certificate is supported for IPsec connections with each configured Server profile. The certificates used is determined by the certificate configured in the profile selected by the user. On the Cube, the client certificate to be used is chosen during the configuration of the server profile (Virtual Server setup). When receiving a certificate, GoSilent Server will verify it against its own trust store, and check the revocation status via OCSP (for IPsec) and CRL (for TLS).**

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.

Evaluator Findings:
The evaluator examined the TSS and confirmed 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 relevant information is found in the following section(s): **TOE Summary Specification section 6.6.2 labeled "FIA_X509_EXT.1/ITT, FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3"** Upon investigation, the evaluator found that the TSS states that: **If the TOE cannot establish a connection to the OCSP server, the TOE will reject the certificate. The certificate of the OCSP responder must include OCSP Signing in its EKU. If the TOE cannot establish a connection to the CDP, the TOE will accept the certificate.**

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.

Evaluator Findings:
The evaluator verified that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator ensured that the guidance documentation contains instructions on how this configuration action is performed. The relevant information is found in the following section(s): **TOE Summary Specification section 6.6.2 labeled "FIA_X509_EXT.1/ITT, FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3"** Upon investigation, the evaluator found that the TSS states that: **The TOE uses X.509 certificates to:** • **provide IPsec connections between GoSilent Server and Cube devices** • **verify the identity of the Syslog server**

Verdict:

PASS.

4.1.6.9.2 FIA_X509_EXT.2 AGD

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.

Evaluator Findings:
The evaluator also ensured that the guidance documentation describes the configuration required in the operating environment so the TOE can use the certificates. The guidance documentation also includes any required configuration on the TOE to use the certificates. The guidance document also describes 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. The relevant information is found in the following section(s): **"Certificate Management"** Upon investigation, the evaluator found that the AGD activity states that: **An external CA certificate and key may be loaded onto the system, or a certificate bundle can be loaded into the TOE via the GUI to provide trust for certificate chains and designate trust anchors:** **On the Cube, navigate to the following:** • **'Device > Certificate Management'** • **Note: During the configuration of the server profile (Virtual Server setup) on the Cube, a drop-down menu is provided to select a certificate to use for the connection.** **On GSS, navigate to the following:** • **'Maintenance > Certificates'** • **Note: When importing a certificate onto GSS, options for "v2" or "WebDashboard" are provided for the administrator to define the certificate usage.**

Verdict:

PASS.

4.1.6.10 FIA_X509_EXT.3 X509 CERTIFICATE REQUESTS (CPP_ND_V3.0E)

4.1.6.10.1 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.

Evaluator Findings:
If the ST author selects "device-specific information", the evaluator verified that the TSS contains a description of the device-specific fields used in certificate requests. The relevant information is found in the following section(s): **TOE Summary Specification section 6.6.2 labeled "FIA_X509_EXT.1/ITT, FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3"** Upon investigation, the evaluator found that the TSS states that: **For a Certificate Signing Request, a Common Name may be specified to include in the certificate.** Other than the common name there is no claim for device specific information.

Verdict:

PASS.

4.1.6.10.2 FIA_X509_EXT.3 AGD

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 the AGD includes instructions for establishing these fields before creating the Certification Request.

Evaluator Findings:
The evaluator checked and ensured that the AGD 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 ensured that the AGD includes instructions for establishing these fields before creating the Certification Request. The relevant information is found in the following section(s): **"Certificate Management"** Upon investigation, the evaluator found that the AGD activity states that: **Certificate signing requests can also be generated through the GUI:** **On the Cube, navigate to the following:** • **'Device > Certificate Management > Manage > Create CSR'** • **Key size P-384 or P-521 can be selected from the drop-down menu.** **On GSS, navigate to the following:** • **'Maintenance > Certificates > Create CSR'** • **Key size P-384 or P-521 can be selected from the drop-down menu.** **Administrators may configure both CN and O fields for values that are expected to be present in certificates received from the remote side during IPsec connection setups. The CN value is configurable by default in the**

'System Settings > VPN' menu, however the O value configurability must be enabled using the following steps:

(1) Enable the O value configuration functionality

Log into GSS as an administrator and navigate to 'System Settings > VPN GUI'.

The "Enable Organization for certificate subject name" check box must be selected.

(2) Restart GSS & Cubes

The GSS must be restarted for a change in this setting to take effect. Each Cube must be restarted three (3) times for a change in this setting to take effect:

(i) Triggers updated settings to be downloaded from GSS

(ii) Cube detects the setting change and reconfigure the network functionality.

(iii) Reconfigured networking functionality to take effect on the Cube.

(3) Configure O value for each applicable Cube on GSS

Log in to the GSS as an administrator and navigate to 'System Settings > VPN Clients GUI'. Add a new cube, or edit an existing Cube entry, and configure the Organization value.

(4) Configure the O value for the GSS on each applicable Cube.

Log in to the Cube as an administrator and select the 'Server Profiles' page. Add a new profile or edit an existing profile entry and configure the Organization value.

Note: When generating a CSR, the TOE does not include the 'O' value. This must be manually configured per the instructions above, or is the responsibility of the signing CA to insert the 'O' value into the certificate.

The TOE will check the 'O' value if it is present in the certificate.

Verdict:

PASS.

4.1.7 SECURITY MANAGEMENT (FMT)

4.1.7.1 GENERAL REQUIREMENTS FOR DISTRIBUTED TOES (CPP_ND_V3.0E)

4.1.7.1.1 GENERAL REQUIREMENTS FOR DISTRIBUTED TOES TSS

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

Evaluator Findings:
The evaluator confirmed that all relevant aspects of each TOE component are covered by the FMT SFRs. Please refer to the TSS Assurance Activity findings in the FMT_MOF.1/ManualUpdate, FMT_MTD.1/CoreData, FMT_SMF.1, and FMT_SMF.2 below.

Verdict:

PASS.

4.1.7.1.2 GENERAL REQUIREMENTS FOR DISTRIBUTED TOES AGD

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

Evaluator Findings:
The evaluator confirmed that all relevant aspects of each TOE component are covered by the FMT SFRs. Please refer to the Guidance Assurance Activity findings in the FMT_MOF.1/ManualUpdate, FMT_MTD.1/CoreData, FMT_SMF.1, and FMT_SMF.2 below.

Verdict:

PASS.

4.1.7.2 FMT_MOF.1/MANUALUPDATE MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOUR (CPP_ND_V3.0E)

4.1.7.2.1 FMT_MOF.1/MANUALUPDATE TSS

For distributed TOEs see Section 5.1.4.1. There are no specific requirements for non-distributed TOEs.

4.1.7.2.2 FMT_MOF.1/MANUALUPDATE AGD

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).

Evaluator Findings:
The evaluator examined the guidance documentation and determined that any necessary steps to perform manual update are described. The guidance documentation also provides warnings regarding functions that may cease to operate during the update (if applicable). The relevant information is found in the following section(s): **"Installation and Verification"** Upon investigation, the evaluator found that the AGD activity states that: **Authorized administrators can perform manual updates to the TOE software by connecting to the component in which they wish to update. Updates are downloaded from a specified URI using HTTPS.** **Manual updates to the GSS can be performed by logging into the GSS GUI and navigating to:** **Maintenance > Client Firmware** **Drag and drop the downloaded image file into the area specified.** **Manual updates to the Cube can be performed by logging into the Cube GUI and clicking on the 'Bell' icon in the top right corner of the GUI and then clicking 'Check for Updates'.** **Prior to the installation of a software update, the Cube will tear down any active tunnels before the update is applied to ensure the integrity of the update process. Once the update process has completed successfully on the Cube, the tunnel is reestablished. On GSS, when an update is successfully installed, GSS will reboot automatically, after which the update will take effect.**

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

Evaluator Findings:
The evaluator examined the guidance documentation and ensured that it describes all steps how to update all TOE components. This contains a description of the order in which components need to be updated if the order is relevant to the update process. The guidance documentation also provides warnings regarding functions of TOE components and the overall TOE that may cease to operate during the update (if applicable). The relevant information is found in the following section(s): **"Installation and Verification"** Upon investigation, the evaluator found that the AGD activity states that: **Authorized administrators can perform manual updates to the TOE software by connecting to the component in which they wish to update. Updates are downloaded from a specified URI using HTTPS. All update files are signed with an ID Technologies security key. This signature is checked against the ID Technologies Security Public Key by the TOE upon download and before installation. If the signature check or package upload fails, both the Cube and GSS will not proceed with the installation and instead continue executing the current version. ID Technologies should be contacted for support if subsequent failures occur.** **Manual updates to the GSS can be performed by logging into the GSS GUI and navigating to:** **Maintenance > Client Firmware** **Drag and drop the downloaded image file into the area specified.** **Manual updates to the Cube can be performed by logging into the Cube GUI and clicking on the 'Bell' icon in the top right corner of the GUI and then clicking 'Check for Updates'.** **Prior to the installation of a software update, the Cube will tear down any active tunnels before the update is applied to ensure the integrity of the update process. Once the update process has completed successfully on the Cube, the tunnel is reestablished. On GSS, when an update is successfully installed, GSS will reboot automatically, after which the update will take effect.**

Verdict:

PASS.

4.1.7.3 FMT_MOF.1/SERVICES MANAGEMENT OF SECURITY FUNCTIONS BEHAVIOUR (CPP_ND_V3.0E)

4.1.7.3.1 FMT_MOF.1/SERVICES TSS

For distributed TOEs see Section 2.4.1.1 of the PP.

Evaluator Findings:
Section 2.4.1.1 of the PP specifies the following: For distributed TOEs, the evaluator shall verify that the TSS describes how every function related to security management is realized for every TOE component and shared between different TOE components. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs. The relevant information is found in the following section(s): **TOE Summary Specification section labeled Security Management (FMT)**

Upon investigation, the evaluator found that the TSS states that:

The TSF shall restrict the ability to enable the functions to perform manual updates to Security Administrators.

The TSF shall restrict the ability to start and stop the functions services to Security Administrators.

The TSF shall restrict the ability to manage the TSF data to Security Administrators.

The TSF shall restrict the ability to [[manage]] the [cryptographic keys and certificates used for VPN operation] to [Security Administrators].

The TSF shall be capable of performing the following management functions:

- **Ability to administer the TOE remotely;**
- **Ability to configure the access banner;**
- **Ability to configure the remote session inactivity time before session termination;**
- **Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;**
- **Ability to start and stop services;**
- **Ability to modify the behaviour of the transmission of audit data to an external IT entity;**
- **Ability to manage the cryptographic keys;**
- **Ability to configure the cryptographic functionality;**
- **Ability to configure the lifetime for IPsec SAs;**
- **Ability to configure the list of supported (D)TLS ciphers;**
- **Ability to configure the interaction between TOE components;**
- **Ability to set the time which is used for time-stamps;**
- **Ability to configure the reference identifier for the peer;**
- **Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;**
- **Ability to generate Certificate Signing Request (CSR) and process CA certificate response;**
- **Ability to administer the TOE locally;**
- **Ability to configure the local session inactivity time before session termination or locking;**
- **Ability to configure the authentication failure parameters for FIA_AFL.1;**

The TSF shall maintain the roles:

- **Security Administrator.**

The TSF shall be able to associate users with roles.

The TSF shall ensure that the conditions

- **The Security Administrator role shall be able to administer the TOE remotely are satisfied.**

For non-distributed TOEs, the evaluator shall ensure the TSS lists the services the Security Administrator is able to start and stop and how that how that operation is performed.

Evaluator Findings:
This TOE is distributed, therefore this activity is not applicable.

Verdict:

PASS.

4.1.7.3.2 FMT_MOF.1/SERVICES AGD

For distributed TOEs see Section 2.4.1.2 of the PP.

Evaluator Findings:
Section 2.4.1.2 of the PP specifies the following: For distributed TOEs, the evaluator shall verify that the Guidance Documentation to describe management of each TOE component. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs. The relevant information is found in the following section(s): **"Configuration Guidance"** Upon investigation, the evaluator found that the AGD activity states that: **"Administration Interfaces"** **Only the following administration interfaces may be used:** a) **GoSilent Server GUI. Administrative GUI interface for managing all GoSilent configurations on the server and other GoSilent platforms.** ○ **Local access is provided for the configured whitelisted source IP address of an administrator device, and by navigating to the local IP address of GSS through the (virtual) MGMT Ethernet interface using HTTPS via web browser to TCP port 4439.** ○ **Remote access is provided to administrators connecting from Cube clients (for any client that has a valid GSS administrator account) using HTTPS via web browser to the MGMT IP address of the GSS over TCP port 443.** b) **GoSilent Cube GUI. User GUI interface that provides limited functionality for initial Cube configuration.** ○ **Local access is provided to users by navigating to the local IP address of the cube using HTTPS via web browser over port 4439 or by navigating to the URL https://setup.gosilent.** **Configuration of some Cube functions is done via the GSS GUI. The Cube subsequently downloads these configuration settings from GSS upon connection establishment.** **"Setting Time"** **The GoSilent Server and GoSilent Cube each maintain a system clock used to provide date/time details for use by the TOE.** a) **GoSilent Server receives time information from the virtualization platform. Upon startup, the GSS sets its internal clock to that time from ESXi. Subsequently, the GSS internal clock is used and subsequent synchronization with the ESXi clock is not performed. There is a delay in updating the clock until the next GoSilent Server restart. Administrators may also choose to set the time manually on GSS by navigating to the following:** **Network > Time and Date** b) **GoSilent Cube allows authorized administrators to manually set the time by logging into the Cube GUI and navigating to the following:**

Device > Date and Time Settings

Once displayed values have been modified, click *'Set Date and Time'* to apply the changes.

"Account Lockout"

GSS supports automatic termination of inactive local and remote administrative sessions, in addition to manual session termination by an administrator. To modify these functions and the thresholds of GSS, login to the GSS GUI and navigate to the following:

'*System Settings > Admin Console*'

and scroll down to the following settings for administrator defined configuration:

- ***'Session inactivity logout interval'* (Note: this parameter must not be '0')**

An administrator may terminate their own session by navigating to the user icon in the top right corner of the GUI and selecting '*Logout*'.

The Cube also supports automatic termination of inactive local sessions, in addition to manual session termination. To modify these functions and the thresholds of the Cube, log into the GSS GUI and navigate to the following:

'*System Settings > GoSilent*'

and scroll down to the following settings for administrator defined configuration:

- ***'Session inactivity logout interval'* (Note: this parameter must not be '0')**
- ***'Session inactivity warning interval'* (Note: Must be less than the logout interval if the logout interval is not 0)**

A user may terminate their own session by navigating to the user icon in the top right corner of the GUI and selecting '*Logout*'.

"Login Banner"

To modify the login banner text for GSS, log into the GSS GUI and navigate to the following:

- **'*System Settings > Admin Console*'**

and scroll down to the '*Enable Login Banner*' setting and ensure the box is checked.

Continue scrolling down to the '*Login Banner Text*' setting and enter the desired login banner message to be displayed.

To modify the login banner text for Cube, log into the GSS GUI and navigate to the following:

- **'*System Settings > GoSilent*'**

and scroll down to the '*Enable Login Banner*' setting and ensure the box is checked.

Continue scrolling down to the '*Login Banner Text*' setting and enter the desired login banner message to be displayed.

"Authentication"

To modify the authentication parameters for GSS, log into the GSS GUI and navigate to the following:

- **'*System Settings > Admin Console*'**

and scroll down to the '*Minimum password length*' setting and enter a value. This parameter should be configured to at least 8 characters.

To modify the authentication parameters for Cube, log into the GSS GUI and navigate to the following:

- **'*System Settings > GoSilent*'**

and scroll down to the '*Minimum password length*' setting and enter a value.

This parameter should be configured to at least 8 characters, and up to 40 characters.

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the TSS lists the services the Security Administrator is able to start and stop and how that how that operation is performed.

Evaluator Findings:
This TOE is distributed, therefore this activity is not applicable.

Verdict:

PASS.

4.1.7.4 FMT_MTD.1/COREDATA MANAGEMENT OF TSF DATA (CPP_ND_V3.0E)

4.1.7.4.1 FMT_MTD.1/COREDATA TSS

For each administrative function identified in the guidance documentation that is accessible through an interface prior to administrator log-in, the evaluator shall confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

Evaluator Findings:

The evaluator confirmed that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

The relevant information is found in the following section(s): **TOE Summary Specification section labeled "FMT_MOF.1/ManualUpdate, FMT_MOF.1/Services, FMT_MTD.1/CoreData, FMT_MTD.1/CryptoKeys, FMT_SMF.1, FMT_SMF.1/FFW, FMT_SMF.1/VPN, FMT_SMR"**

Upon investigation, the evaluator found that the TSS states that: **The GoSilent Server provides the following management capabilities to authorized administrators both locally and remotely:**

- **Perform manual updates of GoSilent Server (which includes validation using digital signatures);**
- **Enable and disable audit logging (although note that the evaluated configuration requires auditing to be enabled at all times);**
- **Manage (import) cryptographic keys and certificates, including PSKs for IPsec;**
- **Ability to configure the access banner;**
- **Ability to configure the session inactivity time for local and remote sessions before session termination;**
- **Ability to configure the authentication failure parameters for FIA_AFL.1;**
- **Ability to configure firewall rules;**
- **Ability to configure the cryptographic functionality;**
- **Ability to configure the lifetime for IPsec SAs;**
- **Ability to configure the transmission of audit data, including the ciphersuite used when connecting to the audit server;**
- **Ability to set the time which is used for time-stamps;**
- **Ability to configure the reference identifier for the peer**
- **Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;**
- **Ability to configure packet filtering rules.**

GoSilent Cube provides the following management capabilities to authorized administrators:

- **Perform manual updates of the GoSilent Cube the administrator is connected to (which includes validation using digital signatures);**
- **Manage (import) cryptographic keys and certificates, including PSKs for IPsec;**
- **Ability to configure the cryptographic functionality;**
- **Ability to set the time which is used for time-stamps;**
- **Ability to configure the reference identifier for the peer**
- **Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;**
- **Ability to configure the packet filtering rules.**

The Cube provides the ability to pull additional configuration information from the GoSilent Server it is registered with upon connection establishment and includes the following:

- **Ability to configure the access banner;**
- **Ability to configure the session inactivity time before session termination or locking;**
- **Ability to configure the authentication failure parameters for FIA_AFL.1;**
- **Ability to configure the transmission of audit data;**

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.

Evaluator Findings:
If the TOE supports handling of X.509v3 certificates and implements a trust store, the evaluator examined the TSS and determined that it contains sufficient information to describe how the ability to manage the TOE's trust store is restricted. The relevant information is found in the following section(s): **TOE Summary Specification FMT_MOF.1/ManualUpdate, FMT_MOF.1/Services, FMT_MTD.1/CoreData, FMT_MTD.1/CryptoKeys, FMT_SMF.1, FMT_SMF.1/FFW, FMT_SMF.1/VPN, FMT_SMR.2** Upon investigation, the evaluator found that the TSS states that: **The GoSilent Server provides the following management capabilities to authorized administrators both locally and remotely:** • **Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;**

Verdict:

PASS.

4.1.7.4.2 FMT_MTD.1/COREDATA AGD

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.

Evaluator Findings:

The evaluator reviewed the guidance documentation and determined 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.

The relevant information is found in the following section(s): **"Configuration Guidance"**

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

"Administration Interfaces"

Only the following administration interfaces may be used:

c) **GoSilent Server GUI. Administrative GUI interface for managing all GoSilent configurations on the server and other GoSilent platforms.**

- **Local access is provided for the configured whitelisted source IP address of an administrator device, and by navigating to the local IP address of GSS through the (virtual) MGMT Ethernet interface using HTTPS via web browser to TCP port 4439.**
- **Remote access is provided to administrators connecting from Cube clients (for any client that has a valid GSS administrator account) using HTTPS via web browser to the MGMT IP address of the GSS over TCP port 443.**

d) **GoSilent Cube GUI. User GUI interface that provides limited functionality for initial Cube configuration.**

- **Local access is provided to users by navigating to the local IP address of the cube using HTTPS via web browser over port 4439 or by navigating to the URL https://setup.gosilent.**

Configuration of some Cube functions is done via the GSS GUI. The Cube subsequently downloads these configuration settings from GSS upon connection establishment.

"Setting Time"

The GoSilent Server and GoSilent Cube each maintain a system clock used to provide date/time details for use by the TOE.

c) **GoSilent Server receives time information from the virtualization platform. Upon startup, the GSS sets its internal clock to that time from ESXi. Subsequently, the GSS internal clock is used and subsequent synchronization with the ESXi clock is not performed. There is a delay in updating the clock until the next GoSilent Server restart. Administrators may also choose to set the time manually on GSS by navigating to the following:**

Network > Time and Date

d) **GoSilent Cube allows authorized administrators to manually set the time by logging into the Cube GUI and navigating to the following:**

Device > Date and Time Settings

Once displayed values have been modified, click *'Set Date and Time'* to apply the changes.

"Account Lockout"

GSS supports automatic termination of inactive local and remote administrative sessions, in addition to manual session termination by an administrator. To modify these functions and the thresholds of GSS, login to the GSS GUI and navigate to the following:

'*System Settings > Admin Console*'

and scroll down to the following settings for administrator defined configuration:

- ***'Session inactivity logout interval'* (Note: this parameter must not be '0')**

An administrator may terminate their own session by navigating to the user icon in the top right corner of the GUI and selecting '*Logout*'.

The Cube also supports automatic termination of inactive local sessions, in addition to manual session termination. To modify these functions and the thresholds of the Cube, log into the GSS GUI and navigate to the following:

'*System Settings > GoSilent*'

and scroll down to the following settings for administrator defined configuration:

- ***'Session inactivity logout interval'* (Note: this parameter must not be '0')**
- ***'Session inactivity warning interval'* (Note: Must be less than the logout interval if the logout interval is not 0)**

A user may terminate their own session by navigating to the user icon in the top right corner of the GUI and selecting '*Logout*'.

"Login Banner"

To modify the login banner text for GSS, log into the GSS GUI and navigate to the following:

- **'*System Settings > Admin Console*'**

and scroll down to the '*Enable Login Banner*' setting and ensure the box is checked.

Continue scrolling down to the '*Login Banner Text*' setting and enter the desired login banner message to be displayed.

To modify the login banner text for Cube, log into the GSS GUI and navigate to the following:

- **'*System Settings > GoSilent*'**

and scroll down to the '*Enable Login Banner*' setting and ensure the box is checked.

Continue scrolling down to the '*Login Banner Text*' setting and enter the desired login banner message to be displayed.

"Authentication"

To modify the authentication parameters for GSS, log into the GSS GUI and navigate to the following:

- **'*System Settings > Admin Console*'**

and scroll down to the '*Minimum password length*' setting and enter a value. This parameter should be configured to at least 8 characters.

To modify the authentication parameters for Cube, log into the GSS GUI and navigate to the following:

- **'*System Settings > GoSilent*'**

and scroll down to the '*Minimum password length*' setting and enter a value.

This parameter should be configured to at least 8 characters, and up to 40 characters.

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.

Evaluator Findings:
If the TOE supports handling of X.509v3 certificates and provides a trust store, the evaluator reviewed the guidance documentation **"Certificate Management"** and determined 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 reviewed the guidance documentation and determined that it provides sufficient information for the administrator to securely load CA certificates into the trust store. The evaluator also reviewed the guidance documentation and determined that it explains how to designate a CA certificate a trust anchor. The relevant information is found in the following section(s): **"Certificate Management"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE leverages X509 certificates to provide IPsec connections between GSS and Cube devices, and to verify the identity of the Syslog server.** **During the enablement process for a Cube, the X509 certificate for GSS and the specific Cube must be communicated out of band and imported by an authorized administrator for each component. This communication should be done via secure communication channel.** **The TOE performs several certificate validity checks on upload, and during TLS and IPsec connection establishment. These checks include a check against its own trust store, the expiration date, revocation status via OCSP (IPSec) and CRL (TLS) and verifies the correct extendedKeyUsage purpose set (Server Authentication). During a certificate upload, the TOE also verifies that the certificate chain terminates with a trusted CA certificate. When validating a certificate used for IPsec, the reference identifier will be matched against the DN (See below for configuration). Certificates used for TLS connections will fall back to the CN if the SAN is not present.** **The TOE supports ECDSA key pair generation using P-384 and P-521 curves which is available through the TOE GUI. When generating certificate signing requests, the option is given to display the CSR information on screen to copy or can be downloaded in PEM format. Keys generated during this process are stored on the TOE in non-volatile memory and are not accessible by any interface. Keys are overwritten when a factory reset is performed.** **The TOE supports ECDSA key establishment using P-384 and P-521 curves.** **An external CA certificate and key may be loaded onto the system, or a certificate bundle can be loaded into the TOE via the GUI to provide trust for certificate chains and designate trust anchors:** **On the Cube, navigate to the following:** ○ **'Device > Certificate Management'** ○ **Note: During the configuration of the server profile (Virtual Server setup) on the Cube, a drop-down menu is provided to select a certificate to use for the connection.** **On GSS, navigate to the following:** ○ **'Maintenance > Certificates'** ○ **Note: When importing a certificate onto GSS, options for "v2" or "WebDashboard" are provided for the administrator to define the certificate usage.**

Verdict:

PASS.

4.1.7.5 FMT_MTD.1/CRYPTOKEYS MANAGEMENT OF TSF DATA (CPP_ND_V3.0E)

4.1.7.5.1 FMT_MTD.1/CRYPTOKEYS TSS

For distributed TOEs see Section 2.4.1.1 of the PP.

Evaluator Findings:
For distributed TOEs see Section 2.4.1.1 of the PP.

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.

Evaluator Findings:
The TOE is distributed. See above.

Verdict:

PASS.

4.1.7.5.2 FMT_MTD.1/CRYPTOKEYS AGD

For distributed TOEs see Section 2.4.1.2 of the PP.

Evaluator Findings:
For distributed TOEs see Section 2.4.1.2 of the PP.

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation describes how the operations are performed on the keys the Security Administrator is able to manage.

Evaluator Findings:
This TOE is distributed, therefore this activity is not applicable.

Verdict:

PASS.

4.1.7.6 FMT_SMF.1 SPECIFICATION OF MANAGEMENT FUNCTIONS (CPP_ND_V3.0E)

4.1.7.6.1 FMT_SMF.1 TSS (CONTAINING ALSO REQUIREMENTS ON GUIDANCE DOCUMENTATION AND TESTS)

The evaluator shall examine the TSS, the 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.

Evaluator Findings:

The evaluator examined the TSS and the Guidance Documentation and the TOE as observed during all other testing and confirmed that the management functions specified in FMT_SMF.1 are provided by the TOE.

The relevant information is found in the following TSS and AGD section(s): **TOE Summary Specification section 6.7.1 FMT_MOF.1/ManualUpdate, FMT_MOF.1/Services, FMT_MTD.1/CoreData , FMT_MTD.1/CryptoKeys, FMT_SMF.1, FMT_SMF.1/FFW, FMT_SMF.1/VPN, FMT_SMR.2** and **"Administration Interfaces"**

Upon investigation, the evaluator found that the TSS states that: **Management of the TOE is primarily performed through GoSilent Server. However, initial management of GoSilent Cube, providing enough configuration information for it to connect to GoSilent Server, is required. The TOE supports the role of Security Administrator. The TOE can be administered both locally and remotely.**

The GoSilent Server provides the following management capabilities to authorized administrators both locally and remotely:

- **Perform manual updates of GoSilent Server (which includes validation using digital signatures);**
- **Enable and disable audit logging (although note that the evaluated configuration requires auditing to be enabled at all times);**
- **Manage (import) cryptographic keys and certificates, including PSKs for IPsec;**
- **Ability to configure the access banner;**
- **Ability to configure the session inactivity time for local and remote sessions before session termination;**
- **Ability to configure the authentication failure parameters for FIA_AFL.1;**
- **Ability to configure firewall rules;**
- **Ability to configure the cryptographic functionality;**
- **Ability to configure the lifetime for IPsec SAs;**
- **Ability to configure the transmission of audit data, including the ciphersuite used when connecting to the audit server;;**
- **Ability to set the time which is used for time-stamps;**
- **Ability to configure the reference identifier for the peer**
- **Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;**
- **Ability to configure the packet filtering rules**

GoSilent Cube provides the following management capabilities to authorized administrators locally:

- **Perform manual updates of the GoSilent Cube the administrator is connected to (which includes validation using digital signatures);**
- **Manage (import) cryptographic keys and certificates, including PSKs for IPsec;**
- **Ability to configure the cryptographic functionality;**
- **Ability to set the time which is used for time-stamps;**
- **Ability to configure the reference identifier for the peer**
- **Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;**
- **Ability to configure the packet filtering rules**

The Cube provides the ability to pull additional configuration information from the GoSilent Server it is registered with upon connection establishment and includes the following:

- **Ability to configure the access banner;**
- **Ability to configure the session inactivity time before session termination or locking;**
- **Ability to configure the authentication failure parameters for FIA_AFL.1;**

- **Ability to configure the transmission of audit data;**

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

Only the following administration interfaces may be used:

e) **GoSilent Server GUI. Administrative GUI interface for managing all GoSilent configurations on the server and other GoSilent platforms.**
   - **Local access is provided for the configured whitelisted source IP address of an administrator device, and by navigating to the local IP address of GSS through the (virtual) MGMT Ethernet interface using HTTPS via web browser to TCP port 4439.**
   - **Remote access is provided to administrators connecting from Cube clients (for any client that has a valid GSS administrator account) using HTTPS via web browser to the MGMT IP address of the GSS over TCP port 443.**

f) **GoSilent Cube GUI. User GUI interface that provides limited functionality for initial Cube configuration.**
   - **Local access is provided to users by navigating to the local IP address of the cube using HTTPS via web browser over port 4439 or by navigating to the URL https://setup.gosilent.**

Configuration of some Cube functions is done via the GSS GUI. The Cube subsequently downloads these configuration settings from GSS upon connection establishment.

The evaluator shall confirm that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface).

Evaluator Findings:

The evaluator confirmed that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface).

The relevant information is found in the following section(s): **TOE Summary Specification section 6.7.1 FMT_MOF.1/ManualUpdate, FMT_MOF.1/Services, FMT_MTD.1/CoreData, FMT_MTD.1/CryptoKeys, FMT_SMF.1, FMT_SMF.1/FFW, FMT_SMF.1/VPN, FMT_SMR.2**

Upon investigation, the evaluator found that the TSS states that: **The GoSilent Server provides the following management capabilities to authorized administrators both locally and remotely:**

- **Perform manual updates of GoSilent Server (which includes validation using digital signatures);**
- **Enable and disable audit logging (although note that the evaluated configuration requires auditing to be enabled at all times);**
- **Manage (import) cryptographic keys and certificates, including PSKs for IPsec;**
- **Ability to configure the access banner;**
- **Ability to configure the session inactivity time for local and remote sessions before session termination;**
- **Ability to configure the authentication failure parameters for FIA_AFL.1;**
- **Ability to configure firewall rules;**
- **Ability to configure the cryptographic functionality;**
- **Ability to configure the lifetime for IPsec SAs;**
- **Ability to configure the transmission of audit data, including the ciphersuite used when connecting to the audit server;**
- **Ability to set the time which is used for time-stamps;**
- **Ability to configure the reference identifier for the peer**
- **Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;**
- **Ability to configure packet filtering rules.**

GoSilent Cube provides the following management capabilities to authorized administrators:

- **Perform manual updates of the GoSilent Cube the administrator is connected to (which includes validation using digital signatures);**
- **Manage (import) cryptographic keys and certificates, including PSKs for IPsec;**
- **Ability to configure the cryptographic functionality;**
- **Ability to set the time which is used for time-stamps;**
- **Ability to configure the reference identifier for the peer**
- **Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;**
- **Ability to configure the packet filtering rules.**

The Cube provides the ability to pull additional configuration information from the GoSilent Server it is registered with upon connection establishment and includes the following:

- **Ability to configure the access banner;**

- **Ability to configure the session inactivity time before session termination or locking;**
- **Ability to configure the authentication failure parameters for FIA_AFL.1;**
- **Ability to configure the transmission of audit data;**

The evaluator shall examine the TSS and the Guidance Documentation to verify they both describe the local administrative interface.

Evaluator Findings:

The evaluator examined the TSS and the Guidance Documentation to verify they both describe the local administrative interface.

The relevant information is found in the following section(s): **TOE Summary Specification section 6.7.1 FMT_MOF.1/ManualUpdate, FMT_MOF.1/Services, FMT_MTD.1/CoreData , FMT_MTD.1/CryptoKeys, FMT_SMF.1, FMT_SMF.1/FFW, FMT_SMF.1/VPN, FMT_SMR.2**

Upon investigation, the evaluator found that the TSS states that: **The GoSilent Server provides the following management capabilities to authorized administrators both locally and remotely:**

- **Perform manual updates of GoSilent Server (which includes validation using digital signatures);**
- **Enable and disable audit logging (although note that the evaluated configuration requires auditing to be enabled at all times);**
- **Manage (import) cryptographic keys and certificates, including PSKs for IPsec;**
- **Ability to configure the access banner;**
- **Ability to configure the session inactivity time for local and remote sessions before session termination;**
- **Ability to configure the authentication failure parameters for FIA_AFL.1;**
- **Ability to configure firewall rules;**
- **Ability to configure the cryptographic functionality;**
- **Ability to configure the lifetime for IPsec SAs;**
- **Ability to configure the transmission of audit data, including the ciphersuite used when connecting to the audit server;**
- **Ability to set the time which is used for time-stamps;**
- **Ability to configure the reference identifier for the peer**
- **Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;**
- **Ability to configure packet filtering rules.**

GoSilent Cube provides the following management capabilities to authorized administrators:

- **Perform manual updates of the GoSilent Cube the administrator is connected to (which includes validation using digital signatures);**
- **Manage (import) cryptographic keys and certificates, including PSKs for IPsec;**
- **Ability to configure the cryptographic functionality;**
- **Ability to set the time which is used for time-stamps;**
- **Ability to configure the reference identifier for the peer**
- **Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;**
- **Ability to configure the packet filtering rules.**

The Cube provides the ability to pull additional configuration information from the GoSilent Server it is registered with upon connection establishment and includes the following:

- **Ability to configure the access banner;**
- **Ability to configure the session inactivity time before session termination or locking;**
- **Ability to configure the authentication failure parameters for FIA_AFL.1;**
- **Ability to configure the transmission of audit data;**

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

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

Only the following administration interfaces may be used:

g) **GoSilent Server GUI. Administrative GUI interface for managing all GoSilent configurations on the server and other GoSilent platforms.**
   - **Local access is provided for the configured whitelisted source IP address of an administrator device, and by navigating to the local IP address of GSS through the (virtual) MGMT Ethernet interface using HTTPS via web browser to TCP port 4439.**
   - **Remote access is provided to administrators connecting from Cube clients (for any client that has a valid GSS administrator account) using HTTPS via web browser to the MGMT IP address of the GSS over TCP port 443.**

h) **GoSilent Cube GUI. User GUI interface that provides limited functionality for initial Cube configuration.**
   - **Local access is provided to users by navigating to the local IP address of the cube using HTTPS via web browser over port 4439 or by navigating to the URL https://setup.gosilent.**

Configuration of some Cube functions is done via the GSS GUI. The Cube subsequently downloads these configuration settings from GSS upon connection establishment.

The evaluator shall ensure the Guidance Documentation includes appropriate warnings for the administrator to ensure the interface is local.

Evaluator Findings:
The evaluator ensured the Guidance Documentation includes appropriate warnings for the administrator to ensure the interface is local. The relevant information is found in the following section(s): **Administration Interfaces** Upon investigation, the evaluator found that the AGD activity states that: **Only the following administration interfaces may be used:** a) **GoSilent Server GUI. Administrative GUI interface for managing all GoSilent configurations on the server and other GoSilent platforms.** ○ **Local access is provided for the configured whitelisted source IP address of an administrator device, and by navigating to the local IP address of GSS through the (virtual) MGMT Ethernet interface using HTTPS via web browser to TCP port 4439.** ○ **Remote access is provided to administrators connecting from Cube clients (for any client that has a valid GSS administrator account) using HTTPS via web browser to the MGMT IP address of the GSS over TCP port 443.** b) **GoSilent Cube GUI. User GUI interface that provides limited functionality for initial Cube configuration.** ○ **Local access is provided to users by navigating to the local IP address of the cube using HTTPS via web browser over port 4439 or by navigating to the URL https://setup.gosilent.** **Configuration of some Cube functions is done via the GSS GUI. The Cube subsequently downloads these configuration settings from GSS upon connection establishment.**

For distributed TOEs with the option 'ability to configure the interaction between TOE components' the evaluator shall examine that the ways to configure the interaction between TOE components is detailed in the TSS and the Guidance Documentation.

Evaluator Findings:
The evaluator examined the TSS and ensured that the ways to configure the interaction between TOE components is detailed in the TSS and the Guidance Documentation. The relevant information is found in the following section(s): **TOE Summary Specification section 6.7.1 "FMT_MOF.1/ManualUpdate, FMT_MOF.1/Services, FMT_MTD.1/CoreData, FMT_MTD.1/CryptoKeys, FMT_SMF.1, FMT_SMF.1/FFW, FMT_SMF.1/VPN, FMT_SMR.2"** Upon investigation, the evaluator found that the TSS states that: **Management of the TOE is primarily performed through GoSilent Server. However, initial management of GoSilent Cube, providing enough configuration information for it to connect to GoSilent Server, is required.** **The Cube provides the ability to pull additional configuration information from the GoSilent Server it is registered with upon connection establishment and includes the following:** • **Ability to configure the access banner;** • **Ability to configure the session inactivity time before session termination or locking;** • **Ability to configure the authentication failure parameters for FIA_AFL.1;** • **Ability to configure the transmission of audit data;** The relevant information is found in the following AGD section(s): **"Installation and Verification"** Upon investigation, the evaluator found that the AGD activity states that: **Each GoSilent Cube must be registered on the GoSilent Server. Registration involves several steps to be completed by an administrator prior to a Cube establishing a connection to GSS. These steps can be completed by navigating to the 'VPN Clients' tab in GSS and clicking on the 'Activate' button next to a listed cube that has been licensed for GSS. The 'Add A Client' window will appear where the 'Device Serial Number' is listed but not configurable. Fill in the 'Client ID' and 'Client Description' fields and click 'Add Client'. Once these steps to prepare the Cube for registration have been completed, a connection attempt from the Cube to GSS can be made.** **Note: The first connection attempt from the Cube to GSS will complete the enablement process and may take longer to establish than subsequent connection attempts.** **The GoSilent Cube is always the initiator for connections to GSS, therefore should the connection be unintentionally broken during the registration process, the user or administrator should check all network connections for the GSS and the Cube to ensure the integrity of the network and then restart the enablement process described above. Should the enablement process not complete successfully, the administrator can remove the Cube in GSS by navigating to 'VPN Clients' and clicking the trash can icon ('Remove') next to the Cube experiencing issues, reboot the Cube device, and restarting the Enablement process again.**

The evaluator shall check that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS and Guidance Documentation.

Evaluator Findings:
The evaluator checked that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS row **FMT_MOF.1/ManualUpdate, FMT_MOF.1/Services, FMT_MTD.1/CoreData, FMT_MTD.1/CryptoKeys, FMT_SMF.1, FMT_SMF.1/FFW, FMT_SMF.1/VPN, FMT_SMR.2** and the Guidance Documentation. This was verified throughout the testing process during the evaluation

(If configure local audit is selected) The evaluator shall examine the TSS and Guidance Documentation to ensure that a description of the logging implementation is described in enough detail to determine how log files are maintained on the TOE.

Evaluator Findings:
N/A. “Configure local audit” is not selected.

Verdict:

PASS.

4.1.7.6.2 FMT_SMF.1 AGD

See Section 4.1.6.6.1.

4.1.7.7 FMT_SMF.1/FFW

The evaluation activities specified for FMT_SMF.1 in the Supporting Document for the Base-PP shall be applied in the same way to the newly added management functions defined in FMT_SMF.1/FFW in the FW Module.

4.1.7.8 FMT_SMF.1/VPN

4.1.7.8.1 FMT_SMF.1/VPN TSS

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

Evaluator Findings:

The evaluator examined the TSS to confirm that all management functions specified in FMT_SMF.1/VPN are provided by the TOE. The relevant information is found in the following section(s): **TOE Summary Specification section "FMT_MOF.1/ManualUpdate, FMT_MOF.1/Services, FMT_MTD.1/CoreData, FMT_MTD.1/CryptoKeys, FMT_SMF.1, FMT_SMF.1/FFW, FMT_SMF.1/VPN, FMT_SMR.2"**

Upon investigation, the evaluator found that the TSS states that: **The GoSilent Server provides the following management capabilities to authorized administrators both locally and remotely:**

- **Perform manual updates of GoSilent Server (which includes validation using digital signatures);**
- **Enable and disable audit logging (although note that the evaluated configuration requires auditing to be enabled at all times);**
- **Manage (import) cryptographic keys and certificates, including PSKs for IPsec;**
- **Ability to configure the access banner;**
- **Ability to configure the session inactivity time for local and remote sessions before session termination;**
- **Ability to configure the authentication failure parameters for FIA_AFL.1;**
- **Ability to configure firewall rules;**
- **Ability to configure the cryptographic functionality;**
- **Ability to configure the lifetime for IPsec SAs;**
- **Ability to configure the transmission of audit data, including the ciphersuite used when connecting to the audit server;**
- **Ability to set the time which is used for time-stamps;**
- **Ability to configure the reference identifier for the peer**
- **Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;**
- **Ability to configure packet filtering rules.**

GoSilent Cube provides the following management capabilities to authorized administrators:

- **Perform manual updates of the GoSilent Cube the administrator is connected to (which includes validation using digital signatures);**
- **Manage (import) cryptographic keys and certificates, including PSKs for IPsec;**
- **Ability to configure the cryptographic functionality;**
- **Ability to set the time which is used for time-stamps;**
- **Ability to configure the reference identifier for the peer**
- **Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;**
- **Ability to configure the packet filtering rules.**

The Cube provides the ability to pull additional configuration information from the GoSilent Server it is registered with upon connection establishment and includes the following:

- **Ability to configure the access banner;**
- **Ability to configure the session inactivity time before session termination or locking;**
- **Ability to configure the authentication failure parameters for FIA_AFL.1;**
- **Ability to configure the transmission of audit data;**

4.1.7.8.2 FMT_SMF.1/VPN AGD

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

Evaluator Findings:
The evaluator examined the AGD and determined that it confirms all management functions specified in FMT_SMF.1/VPN are provided by the TOE. The relevant information is found in the following section(s): **"Firewall Configuration"** Upon investigation, the evaluator found that the AGD states that: **Administrators can define stateful traffic filtering rules based on a distinct interface in addition to the following network protocol fields:** • **ICMPv4 (Type, Code)** • **ICMPv6 (Type, Code)** • **IPv4 (Source address, Destination address, Transport Layer Protocol)** • **IPv6 (Source address, Destination address, Transport Layer Protocol)** • **TCP (Source port, Destination port)** • **UDP (Source port, Destination port)**

4.1.7.9 FMT_SMR.2 RESTRICTIONS ON SECURITY ROLES (CPP_ND_V3.0E)

4.1.7.9.1 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 (e.g. if local administrators and remote administrators have different privileges or if several types of administrators with different privileges are supported by the TOE).

Evaluator Findings:
The evaluator examined the TSS and determined that it details the TOE supported roles and any restrictions of the roles involving administration of the TOE. The relevant information is found in the following section(s): **TOE Summary Specification section 6.7.1 FMT_MOF.1/ManualUpdate, FMT_MOF.1/Services, FMT_MTD.1/CoreData, FMT_MTD.1/CryptoKeys, FMT_SMF.1, FMT_SMF.1/FFW, FMT_SMF.1/VPN, FMT_SMR.2** Upon investigation, the evaluator found that the TSS states that: **The TOE supports the single administrator role of Security Administrator; no other users have elevated privileges**

Verdict:

PASS.

4.1.7.9.2 FMT_SMR.2 AGD

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.

Evaluator Findings:
The evaluator reviewed the AGD **"Administration Interfaces"** and ensured that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration. The relevant information is found in the following section(s): **"Administration Interfaces"** Upon investigation, the evaluator found that the AGD activity states that: **Only the following administration interfaces may be used:** ○ **GoSilent Server GUI. Administrative GUI interface for managing all GoSilent configurations on the server and other GoSilent platforms.** ▪ **Local access is provided for the configured whitelisted source IP address of an administrator device, and by navigating to the local IP address of GSS through the (virtual) MGMT Ethernet interface using HTTPS via web browser to TCP port 4439.** ▪ **Remote access is provided to administrators connecting from Cube clients (for any client that has a valid GSS administrator account) using HTTPS via web browser to the MGMT IP address of the GSS over TCP port 443.** ○ **GoSilent Cube GUI. User GUI interface that provides limited functionality for initial Cube configuration.** ▪ **Local access is provided to users by navigating to the local IP address of the cube using HTTPS via web browser over port 4439 or by navigating to the URL https://setup.gosilent.** ▪ **Configuration of some Cube functions is done via the GSS GUI. The Cube subsequently downloads these configuration settings from GSS upon connection establishment.**

Verdict:

PASS.

4.1.8 PACKET FILTERING (FPF) (MOD_VPNGW_V1.3)

4.1.8.1 FPF_RUL_EXT.1 PACKET FILTERING RULES TSS (MOD_VPNGW_V1.3)

4.1.8.1.1 FPF_RUL_EXT.1.1 TSS (MOD_VPNGW_V1.3)

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

Evaluator Findings:
The evaluator reviewed the TSS, section 6.5.1 labeled **"FFW_RUL_EXT.1 & FPF_RUL_EXT.1"**, and ensured that it provides a description of the TOE's initialization and startup process, which clearly indicates where processing of network packets begins to take place, and provides a discussion that supports the assertion that packets cannot flow during this process.

The TOE performs stateful packet filtering on all packets received from or being sent to the External Network, MGMT interface or WAN interface (excepting IPsec traffic) of the GoSilent Server.

The boot sequence of the TOE aids in establishing the secure domain and preventing tampering or bypass of security functionality. This includes ensuring the packet filtering rules cannot be bypassed during the boot sequence of the TOE. The following steps list the boot sequence for the TOE:

- **BIOS hardware and memory checks**
- **Loading and initialization of the OS**
- **Self-tests including firmware integrity tests are executed**
    - **The init utility is started (mounts file systems, sets up network cards, and generally starts all the processes that usually are run on the system at startup)**
    - **Daemon programs such as Internet Service Daemon (INETD), Syslogd are started; Routing and forwarding tables are initialized**
    - **Application daemons are loaded, enabling access to the GoSilent Server management GUI**
- **Physical interfaces are active**

Once the interfaces are brought up, they will start to receive and send packets based on the current configuration (or not receive or send any packets if they have not been previously configured). Interfaces are brought up only after successful loading of the kernel and daemons, and these interfaces cannot send or receive packets unless previously configured by an administrator. Since the daemon for the management GUI is not loaded until after the kernel and INETD are initialized, no modification to the security attributes can be made by a user or process other than via the management process.

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

Evaluator Findings:
The evaluator reviewed the TSS **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"** and ensured that it includes a narrative that identifies the components (e.g., active entity such as a process or task) involved in processing the network packets and describes the safeguards that would prevent packets flowing through the TOE without applying the ruleset in the event of a component failure. **The TOE applies a uniform policy to the traffic flows to and from all GoSilent Cube users. By default, no traffic is allowed to flow from GoSilent Cube users to the External Network, no traffic is allowed to flow from the External Network to GoSilent Cube users, and no traffic is allowed to flow between GoSilent Server interfaces.**

Verdict:
PASS.

4.1.8.1.2 FPF_RUL_EXT.1.1 AGD (MOD_VPNGW_V1.3)

The operational guidance associated with this requirement is assessed in the subsequent test EAs.

4.1.8.1.3 FPF_RUL_EXT.1.2 (MOD_VPNGW_V1.3)

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

4.1.8.1.4 FPF_RUL_EXT.1.3 (MOD_VPNGW_V1.3)

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

4.1.8.1.5 FPF_RUL_EXT.1.4 TSS (MOD_VPNGW_V1.3)

The evaluator shall verify that the TSS section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1" describes a packet filtering policy that can use the following fields for each identified protocol, and that the RFCs identified for each protocol are supported:

- IPv4 (RFC 791)
  - source address
  - destination address
  - protocol
- IPv6 (RFC 8200)
  - source address
  - destination address
  - next header (protocol)
- TCP (RFC 793)
  - source port
  - destination port
- UDP (RFC 768)
  - source port
  - destination port

Evaluator Findings:
The evaluator reviewed the TSS **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"** and ensured that it describes a packet filtering policy that can use the following fields for each identified protocol, and that the RFCs identified for each protocol are supported: **By default, the TOE allows administrators to define traffic filtering rules based on a distinct interface and the following network protocol fields:** • **ICMPv4 (Type, Code)** • **ICMPv6 (Type, Code)** • **IPv4 (Source address, Destination address, Transport Layer Protocol)** • **IPv6 (Source address, Destination address, Transport Layer Protocol)** • **TCP (Source port, Destination port)** • **UDP (Source port, Destination port)**

The TSF allows the definition of packet filtering rules by using the following network protocols and protocol fields:

- **IPv4 (RFC 791)**
  - **Source address**
  - **Destination address**
  - **Protocol**
- **IPv6 (RFC 8200)**
  - **Source address**
  - **Destination address**
  - **Next header (protocol)**
- **TCP (RFC 793)**
  - **Source port**
  - **Destination port**
- **UDP (RFC 768)**
  - **Source port**
  - **Destination port**

Conformance with the above listed RFC's has been determined through testing and development and as a requirement of this evaluation.

The TOE logs the actions taken on packets when they are processed by the INPUT, OUTPUT, and FORWARD rulesets.

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

Evaluator Findings:

The evaluator reviewed the TSS **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"** and ensured that it describes how conformance with the identified RFCs has been determined by the TOE developer (e.g., third party interoperability testing, protocol compliance testing).

The TSF allows the definition of packet filtering rules by using the following network protocols and protocol fields:

- **IPv4 (RFC 791)**
  - **Source address**
  - **Destination address**
  - **Protocol**
- **IPv6 (RFC 8200)**
  - **Source address**
  - **Destination address**
  - **Next header (protocol)**
- **TCP (RFC 793)**
  - **Source port**
  - **Destination port**
- **UDP (RFC 768)**
  - **Source port**
  - **Destination port**

Conformance with the above listed RFC's has been determined through testing and development and as a requirement of this evaluation.

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

Evaluator Findings:
The evaluator reviewed the TSS **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"** and ensured that each rule identifies the following actions: permit, discard, and log. The TOE logs the actions taken on packets when they are processed by the INPUT, OUTPUT, and FORWARD rulesets packet.

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

Evaluator Findings:
The evaluator reviewed the TSS **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"** and ensured that it identifies all interface types subject to the packet filtering policy and explains how rules are associated with distinct network interfaces. **The TOE performs stateful packet filtering on all packets received from or being sent to the External Network, MGMT interface or WAN interface (excepting IPsec traffic) of the GoSilent Server.**

Verdict:
PASS.

4.1.8.1.6 FPF_RUL_EXT.1.4 AGD (MOD_VPNGW_V1.3)

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

- IPv4 (RFC 791)
  - destination address
  - protocol
- IPv6 (RFC 8200)
  - source address
  - destination address
  - next header (protocol)
- TCP (RFC 793)
  - source port
  - destination port
- UDP (RFC 768)
  - source port

- destination port

Evaluator Findings:

The evaluator checked the AGD **"Firewall Configuration"** and ensured that it identifies the following protocols as being supported and the following attributes as being configurable within packet filtering rules for the associated protocols:

- IPv4 (RFC 791)
  - source address
  - destination address
  - protocol
- IPv6 (RFC 8200)
  - source address
  - destination address
  - next header (protocol)
- TCP (RFC 793)
  - source port
  - destination port
- UDP (RFC 768)
  - source port
  - destination port

The TOE Administrators can define stateful traffic filtering rules based on a distinct interface in addition to the following network protocol fields:

- **ICMPv4 (Type, Code)**
- **ICMPv6 (Type, Code)**
- **IPv4 (Source address, Destination address, Transport Layer Protocol)**
- **IPv6 (Source address, Destination address, Transport Layer Protocol)**
- **TCP (Source port, Destination port)**
- **UDP (Source port, Destination port)**

Packet filtering rules can be configured with an action to accept (permit/allow), reject (discard/deny), ignore, or to pass the packet on to other rules for more processing.

Netfilter allows the definition of packet filtering rules and processes incoming or outgoing traffic according to the following parameters:

- **IPv4 (RFC 791)**
  - **Source Address**
  - **Destination Address**
  - **Protocol**
- **IPv6 (RFC 8200)**
  - **Source Address**
  - **Destination Address**
  - **Next Header (Protocol)**
- **TCP (RFC 793)**
  - **Source Port**
  - **Destination Port**
- **UDP (RFC 768)**

- **Source Port**
- **Destination Port**

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

Evaluator Findings:
The evaluator checked the AGD section **"Firewall Configuration"** and ensured that it indicates that each rule can identify the following actions: permit, discard, and log. The TOE All traffic that correlates to an existing session is permitted, and any traffic not related to a known session is subject to processing by rules that apply flow policies in an administrator-defined order. Rules can be inserted into the ruleset via GUI. Rules are associated to specific interfaces by defining the interface name in the rule. The GUI displays the order of the rules. If no rule matching the traffic exists, the traffic is denied. By default, all traffic from GoSilent Cube users to the External Network, from External Network to Cube users, and between GSS interfaces is denied. A default rule should be applied in the filter table that denies IPv4 and IPv6 source addresses that match the configured IP address on the TOE's interfaces. **Iptables firewall functions are built on the Netfilter framework that is available in the Linux kernel for packet filtering. Netfilter contains tables. These tables contain chains, and chains contain individual rules. If a packet matches any rule, the rule action will be applied on that packet.** **Custom chains can also be created to save rules in. Each chain in the Filter table has a 'policy', which is the default action taken on packets processed by the chain. The following policies are available:** **a) DROP. Drops a packet without informing the client.** **b) REJECT. Drops a packet and informs the sender.** **c) ACCEPT. Allows the packets to pass the firewall**

The evaluator shall verify that the operational guidance explains how rules are associated with distinct network interfaces. The guidance may describe the other protocols contained within the ST (e.g., IPsec, IKE, potentially HTTPS, SSH, and TLS) that are processed by the TOE.

Evaluator Findings:
The evaluator checked the AGD section **"Firewall Configuration"** and ensured that it explains how rules are associated with distinct network interfaces. **The TOE performs stateful packet filtering via iptables on all packets sent and received from the External Network, MGMT, and WAN interfaces and applies a uniform policy on all traffic to and from GoSilent Cube users.**

The evaluator shall ensure that it is made clear what protocols were not considered as part of the TOE evaluation.

Evaluator Findings:
The evaluator checked the AGD section **"Firewall Configuration"** and ensured that it is clear what protocols were not considered as part of the TOE evaluation. **Administrators can define stateful traffic filtering rules based on a distinct interface in addition to the following network protocol fields:** • **ICMPv4 (Type, Code)**

- **ICMPv6 (Type, Code)**
- **IPv4 (Source address, Destination address, Transport Layer Protocol)**
- **IPv6 (Source address, Destination address, Transport Layer Protocol)**
- **TCP (Source port, Destination port)**
- **UDP (Source port, Destination port)**

Verdict:
PASS.

4.1.8.1.7 FPF_RUL_EXT.1.5 TSS (MOD_VPNGW_V1.3)

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

Evaluator Findings:
The evaluator reviewed the TSS, section **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"**, and ensured that it describes the algorithm applied to incoming packets, including the processing of default rules, determination of whether a packet is part of an established session, and application of administrator defined and ordered ruleset. **By default, the TOE allows administrators to define traffic filtering rules based on a distinct interface and the following network protocol fields:** • **ICMPv4 (Type, Code)** • **ICMPv6 (Type, Code)** • **IPv4 (Source address, Destination address, Transport Layer Protocol)** • **IPv6 (Source address, Destination address, Transport Layer Protocol)** • **TCP (Source port, Destination port)** • **UDP (Source port, Destination port)** **The TOE does not implement a mechanism to ensure conflicting rules can't be configured.**

Verdict:
PASS.

4.1.8.1.8 FPF_RUL_EXT.1.5 AGD (MOD_VPNGW_V1.3)

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

Evaluator Findings:
The evaluator checked the AGD section **"Table Chains"** and ensured that it describes how the order of packet filtering rules is determined and provides the necessary instructions so that an administrator can configure the order of rule processing.

The administrator must configure rules as necessary to ensure the TOE drops and is capable of logging packets in all network traffic that meet the following criteria:

Custom chains can also be created to save rules in. Each chain in the Filter table has a 'policy', which is the default action taken on packets processed by the chain. The following policies are available:

a) DROP. Drops a packet without informing the client.

b) REJECT. Drops a packet and informs the sender.

c) ACCEPT. Allows the packets to pass the firewall.

Verdict:
PASS.

4.1.8.1.9 FPF_RUL_EXT.1.6 TSS (MOD_VPNGW_V1.3)

The evaluator shall verify that the TSS describes the process for applying packet filtering rules and also that the behavior (either by default, or as configured by the administrator) is to discard packets when there is no rule match.

Evaluator Findings:
The evaluator reviewed the TSS, **section 6.5.1 labeled "FFW_RUL_EXT.1 & FPF_RUL_EXT.1"**, and ensured that it describes the process for applying packet filtering rules and also that the behavior (either by default, or as configured by the administrator) is to discard packets when there is no rule match. **Traffic for known sessions is permitted to flow. For traffic not associated with known sessions, rules within information flow policies are processed in an administrator-defined order to determine if the traffic should be forwarded. By default, the TOE behavior is to deny packets when there is no rule match.**

The evaluator shall verify the TSS describes when the IPv4 and IPv6 protocols supported by the TOE differ from the full list provided in the RFC Values for IPv4 and IPv6 table.

Evaluator Findings:
N/A. The IPv4 and IPv6 protocols supported by the TOE do not differ from the full list provided in the RFC Values for IPv4 and IPv6.

Verdict:
PASS.

4.1.8.1.10 FPF_RUL_EXT.1.6 AGD (MOD_VPNGW_V1.3)

The evaluator shall verify that the operational guidance describes the behavior if no rules or special conditions apply to the network traffic.

Evaluator Findings:
The evaluator checked the AGD section **"Firewall Configuration"** and ensured that it describes the behavior if no rules or special conditions apply to the network traffic. **All traffic that correlates to an existing session is permitted, and any traffic not related to a known session is subject to processing by rules that apply flow policies in an administrator-defined order. Rules can be inserted into the ruleset via GUI. Rules are associated to specific interfaces by defining the interface name in the rule. The GUI displays the order of the rules. If no rule matching the traffic exists, the traffic is denied. By default, all traffic from GoSilent Cube users to the External Network, from External Network to Cube users, and between GSS interfaces is denied. A default rule should be applied in the filter table that denies IPv4 and IPv6 source addresses that match the configured IP address on the TOE's interfaces.**

If the behavior is configurable, the evaluator shall verify that the operational guidance provides the appropriate instructions to configure the behavior to discard packets with no matching rules.

Evaluator Findings:
The behavior to discard packets with no matching rules is not configurable; hence, this activity is not applicable.

The evaluator shall verify that the operational guidance describes the range of IPv4 and IPv6 protocols supported by the TOE.

Evaluator Findings:
The evaluator checked the AGD section **"Firewall Configuration"** and ensured that it describes the range of IPv4 and IPv6 protocols supported by the TOE. **Administrators can define stateful traffic filtering rules based on a distinct interface in addition to the following network protocol fields:** • **ICMPv4 (Type, Code)** • **ICMPv6 (Type, Code)** • **IPv4 (Source address, Destination address, Transport Layer Protocol)** • **IPv6 (Source address, Destination address, Transport Layer Protocol)** • **TCP (Source port, Destination port)** • **UDP (Source port, Destination port)**

Verdict:

PASS.

4.1.9 PROTECTION OF THE TSF (FPT)

4.1.9.1 FPT_APW_EXT.1 PROTECTION OF ADMINISTRATOR PASSWORDS (CPP_ND_V3.0E)

4.1.9.1.1 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.

Evaluator Findings:
The evaluator examined 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 also detailed 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. The relevant information is found in the following section(s): **TOE Summary Specification section 6.8.1 labeled "FPT_APW_EXT.1 & FPT_SKP_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **The TOE does not provide access to the filesystem or any administrative interface, including those designed specifically for that purpose, that would allow a user or administrator to view plaintext administrative passwords, pre-shared keys, symmetric keys, or private keys and additionally prevents the reading of any plaintext administrative passwords by instead storing their SHA-256 hash values.** **All administrative password hashes, pre-shared keys, symmetric keys, and private keys are stored in non-volatile memory at non-fixed locations that are otherwise not accessible by administrative users.**

Verdict:

PASS.

4.1.9.1.2 FPT_APW_EXT.1 AGD

None.

4.1.9.2 FPT_ITT.1 BASIC INTERNAL TSF DATA TRANSFER PROTECTION (CPP_ND_V3.0E)

4.1.9.2.1 FPT_ITT.1 TSS

The evaluator shall examine the TSS to determine that, for all communications between components of a distributed TOE, each communications mechanism is identified in terms of the allowed protocols for that IT entity.

Evaluator Findings:
The evaluator examined the TSS and determined that, for all communications between components of a distributed TOE, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The relevant information is found in the following section(s): **TOE Summary Specification section 6.8.3 labeled "FPT_ITT.1"** Upon investigation, the evaluator found that the TSS states that: **All communication between the TOE components uses IPsec to protect the traffic.** **IPsec connections are always initiated by GoSilent Cube; GoSilent Server only receives incoming connections from GoSilent Cube devices.**

The evaluator shall also confirm that all protocols listed in the TSS for these inter-component communications are specified and included in the requirements in the ST.

Evaluator Findings:
The evaluator also confirmed that all protocols listed in the TSS for these inter-component communications are specified and included in the requirements in the ST. The relevant information is found in the following section(s): **TOE Summary Specification section 6.8.3 labeled "FPT_ITT.1"** Upon investigation, the evaluator found that the TSS states that: **IPsec connections are always initiated by GoSilent Cube; GoSilent Server only receives incoming connections from GoSilent Cube devices.**

Verdict:

PASS.

4.1.9.2.2 FPT_ITT.1 AGD

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

Evaluator Findings:
The evaluator confirmed that the guidance documentation contains instructions for establishing the relevant allowed communication channels and protocols between each pair of authorized TOE components, and that it contains recovery instructions should a connection be unintentionally broken. The relevant information is found in the following section(s): **"VPN"** & **"Network"** Upon investigation, the evaluator found that the AGD activity states that: Steps for enabling Cubes authorized for connections to GSS can be found in **"Installation and Verification."** **"Virtual Server Configuration" continues this information and states: The GoSilent Cube always initiates the connection to establish a tunnel with GSS. Before the Cube can initiate this connection, a Virtual Server**

profile containing the GSS server information must be configured on the Cube. This can be done by logging into the Cube by navigating to the following:

Settings > Server Profiles

Click the 'Add' button next to 'Virtual Server' and enter the details for GSS.

Click 'Add Virtual Server'. The newly configured server profile for GSS will be shown in the Server Profiles list.

Note: Ensure that all GSS VPN configurations and Cube enablement have been completed as described above.

Click the 'Connect' button on the GSS server profile. In the 'Are you sure?' window click 'Yes, I'm Sure'.

The GSS server profile will show the status 'Connected'

The 'GoSilent Status' shown in the top left of the Cube GUI will show 'VPN Connected'.

The maximum number of retry attempts that will be made by the Network Watchdog service to re-establish the internet or WAN connection should it fail, or otherwise be unintentionally broken, is configurable (in attempts) to between 5 and 200, with a default configuration of 5 attempts. When this threshold is met, the Cube can be rebooted to initiate the connection again. This figure can be configured by navigating to the following:

'System Settings > GoSilent > Network watcher WAN retries'.

Verdict:

PASS.

4.1.9.3 FPT_SKP_EXT.1 PROTECTION OF TSF DATA (FOR READING OF ALL SYMMETRIC KEYS) (CPP_ND_V3.0E)

4.1.9.3.1 FPT_SKP_EXT.1 TSS

The evaluator shall examine the TSS to determine that it details how any preshared 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.

Evaluator Findings:
The evaluator examined the TSS and determined 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 describes how they are protected/obscured. The relevant information is found in the following section(s): **TOE Summary Specification section 6.8.1 labeled "FPT_APW_EXT.1 & FPT_SKP_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **The TOE does not provide access to the filesystem or any administrative interface, including those designed specifically for that purpose, that would allow a user or administrator to view plaintext administrative passwords, pre-shared keys, symmetric keys, or private keys and additionally prevents the reading of any plaintext administrative passwords by instead storing their SHA-256 hash values.** **All administrative password hashes, pre-shared keys, symmetric keys, and private keys are stored in non-volatile memory at non-fixed locations that are otherwise not accessible by administrative users.**

Verdict:

PASS.

4.1.9.3.2 FPT_SKP_EXT.1 AGD

None.

4.1.9.4 FPT_STM_EXT.1 RELIABLE TIME STAMPS (CPP_ND_V3.0E)

4.1.9.4.1 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.

Evaluator Findings:
The evaluator examined the TSS and ensured 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. The relevant information is found in the following section(s): **TOE Summary Specification section 6.8.5 labeled "FPT_STM_EXT.1** Upon investigation, the evaluator found that the TSS states that: **GoSilent Server and GoSilent Cube each maintain a system clock used to provide date and time information for TOE operations, including reliable timestamping for audit logs, and validation of certificate expiration dates. The time can be manually set by authorized administrators only.** **GoSilent Server receives time information from the virtualization platform to maintain the time across system reboots. When a VM is started, the virtualization system (ESXi or KVM) provides the VM with the virtualization system's internal time value. Upon startup, the GoSilent Server sets its internal clock to that time from the virtualization system. Subsequently, the GoSilent Server internal clock is used and subsequent synchronization with the virtualization system clock is not performed. There is a delay in updating the clock until the next GoSilent Server restart. Time information obtained from the underlying virtualization platform is considered to be reliable because access to the virtualization platform, including the time setting function, is restricted to authorized administrators of the platform only. When setting the time manually on GoSilent Server, only authorized administrators have access to and can modify that function.** **For GoSilent Cube, guidance directs the administrator to manually set the clock on each reboot since it does not include a real-time clock.**

If "obtain time from the underlying virtualization system" is selected, the evaluator shall examine the TSS to ensure that it identifies the VS interface the TOE uses to obtain time. If there is a delay between updates to the time on the VS and updating the time on the TOE, the TSS shall identify the maximum possible delay.

Evaluator Findings:
If “obtain time from the underlying virtualization system” is selected, the evaluator examined the TSS and ensured that it identifies the VS interface the TOE uses to obtain time. If there is a delay between updates to the time on the VS and updating the time on the TOE, the TSS shall identify the maximum possible delay. The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification section 6.8.5 labeled “FPT_STM_EXT.1** Upon investigation, the evaluator found that the TSS states that: **GoSilent Server receives time information from the virtualization platform to maintain the time across system reboots. When a VM is started, the virtualization system (ESXi or KVM) provides the VM with the virtualization system’s internal time value. Upon startup, the GoSilent Server sets its internal clock to that time from the virtualization system. Subsequently, the GoSilent Server internal clock is used and subsequent synchronization with the virtualization system clock is not performed. There is a delay in updating the clock until the next GoSilent Server restart. Time information obtained from the underlying virtualization platform is considered to be reliable because access to the virtualization platform, including the time setting function, is restricted to authorized administrators of the platform only. When setting the time manually on GoSilent Server, only authorized administrators have access to and can modify that function.**

Verdict:

PASS.

4.1.9.4.2 FPT_STM_EXT.1 AGD

The evaluator shall examine 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.

Evaluator Findings:
The evaluator examined the guidance documentation and ensured that 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. The relevant information is found in the following section(s): **"Setting Time"** Upon investigation, the evaluator found that the AGD activity states that: **The GoSilent Server and GoSilent Cube each maintain a system clock used to provide date/time details for use by the TOE.** e) **GoSilent Server receives time information from the virtualization platform. Upon startup, the GSS sets its internal clock to that time from ESXi. Subsequently, the GSS internal clock is used and subsequent synchronization with the ESXi clock is not performed. There is a delay in updating the clock until the next GoSilent Server restart. Administrators may also choose to set the time manually on GSS by navigating to the following:** **Network > Time and Date** f) **GoSilent Cube allows authorized administrators to manually set the time by logging into the Cube GUI and navigating to the following:** **Device > Date and Time Settings** **Once displayed values have been modified, click 'Set Date and Time' to apply the changes.**

If the TOE supports obtaining time from the underlying VS, the evaluator shall verify the Guidance Documentation specifies any configuration steps necessary. If no configuration is necessary, no statement is necessary in the Guidance Documentation. If there is a delay between updates to the time on the VS and updating the time on the TOE, the evaluator shall ensure the Guidance Documentation informs the administrator of the maximum possible delay.

Evaluator Findings:
The AGD does not detail any steps necessary to configure the TOE to obtain time from the underlying VS. The GoSlient Server sets its internal clock from the underlying VS upon startup. The maximum possible delay is the next restart.

Verdict:

PASS.

4.1.9.5 FPT_TST_EXT.1 TSF TESTING (CPP_ND_V3.0E)

4.1.9.5.1 FPT_TST_EXT.1 TSS **[TD0836]**

The evaluator shall examine the TSS to ensure that it details each of the self-tests that are identified by the SFR; 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).

Evaluator Findings:
The evaluator examined the TSS and ensured that it details each of the self-tests that are identified by the SFR; 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" is used). The relevant information is found in the following section(s): **TOE Summary Specification section 6.8.2 labeled "FPT_FLS.1/SelfTest, FPT_TST_EXT.1, FPT_TST_EXT.3"** Upon investigation, the evaluator found that the TSS states that: **At power-on tests are performed on each component to confirm the integrity of the firmware and a statistical assessment of the entropy source. The integrity of the firmware is tested using the cryptographic services described in FCS_COP.1/SigGen, which covers all of the executable code. The noise source health test performs a statistical assessment of 1000 samples. DRBG randomness test continuously compares the current and previously generated blocks to see if they are the same. Identical blocks indicate a failure of the generation function and produce an error and an error return value.**

The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

Evaluator Findings:
The evaluator ensured that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly. The relevant information is found in the following section(s):): **TOE Summary Specification section 6.8.2 labeled "FPT_FLS.1/SelfTest, FPT_TST_EXT.1, FPT_TST_EXT.3"** Upon investigation, the evaluator found that the TSS states that: **If any test fails, that component does not enter an operational state and the network interfaces are not activated. For GoSilent Server, an error is displayed on the local console accessible via ESXi or KVM. For GoSilent Cube, the LEDs indicate the system is not operational. In the event of an initial start-up failure the red and green status LEDs will blink in unison, in the event of a cryptographic service provider failure the red power LED will remain on, and the green status LED will be extinguished.**

If more than one failure response is listed in FPT_TST_EXT.1.2, the evaluator shall examine the TSS to ensure it clarifies which response is associated with which type of failure.

Evaluator Findings:
If more than one failure response is listed in FPT_TST_EXT.1.2, the evaluator shall examine the TSS to ensure it clarifies which response is associated with which type of failure. The relevant information is found in the following section(s):): **TOE Summary Specification section 6.8.2 labeled "FPT_FLS.1/SelfTest, FPT_TST_EXT.1, FPT_TST_EXT.3"** Upon investigation, the evaluator found that the TSS states that: **If any test fails, that component does not enter an operational state and the network interfaces are not activated. For GoSilent Server, an error is displayed on the local console accessible via ESXi or KVM. For GoSilent Cube, the LEDs indicate the system is not operational. In the event of an initial start-up failure the red and green status LEDs will blink in unison; in the event of a DRBG randomness test failure the red power LED will remain on, and the green status LED will be extinguished.** **These tests are sufficient to demonstrate the TSF is operating correctly, as they confirm the integrity of all firmware modules prior to their execution, thereby confirming the modules have not been modified or replaced in any unauthorized manner, and they ensure the DRBG continues to operate successfully providing sufficient entropy in response to any requests.**

For distributed TOEs the evaluator shall examine the TSS to ensure that it details which TOE component performs which self-tests and when these self-tests are run. The evaluator shall also examine the TSS to ensure it describes how the TOE reacts if one or more TOE components fail self-testing (e.g. halting and displaying an error message; failover behaviour).

Evaluator Findings:
The evaluator examined the TSS and ensured that it details which TOE component performs which self-tests and when these self- tests are run. The relevant information is found in the following section(s): **TOE Summary Specification): TOE Summary Specification section 6.8.2 labeled "FPT_FLS.1/SelfTest, FPT_TST_EXT.1, FPT_TST_EXT.3"** Upon investigation, the evaluator found that the TSS states that: **If any test fails, that component does not enter an operational state and the network interfaces are not activated. For GoSilent Server, an error is displayed on the local console accessible via ESXi or KVM. For GoSilent Cube, the LEDs indicate the system is not operational. In the event of an initial start-up failure the red and green status LEDs will blink in unison; in the event of a DRBG randomness test failure the red power LED will remain on, and the green status LED will be extinguished.**

Verdict:

PASS.

4.1.9.5.2 FPT_TST_EXT.1 AGD

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.

Evaluator Findings:
The evaluator also ensured 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 correspond to those described in the TSS. The relevant information is found in the following section(s): **"Firmware Integrity"** Upon investigation, the evaluator found that the AGD activity states that: **On start-up, a firmware integrity test and statistical assessment of the entropy source are executed on both GSS and Cube. If any of these tests fail or otherwise do not complete successfully, the component will not enter an operational state and the network interfaces are not activated. On GSS, a relevant error message is displayed on the local console (accessible via ESXi or KVM interface). On Cube, onboard LEDs indicate the error state and that the system is non-operational.** **For the rectification of error states, diagnostic information displayed in relevant error messages on GSS and behaviour of the LEDs on Cube should be referenced. Administrators should contact ID Technologies in the event any component fails the power-up tests.**

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

Evaluator Findings:
The evaluator examined the guidance documentation and ensured that it describes how to determine from an error message returned which TOE component has failed the self-test. The relevant information is found in the following section(s): **"Firmware Integrity"** Upon investigation, the evaluator found that the AGD activity states that: **On start-up, a firmware integrity test and statistical assessment of the entropy source are executed on both GSS and Cube.** **If any of these tests fail or otherwise do not complete successfully, the component will not enter an operational state and the network interfaces are not activated. On GSS, a relevant error message is displayed on the local console (accessible via ESXi). On Cube, onboard LEDs indicate the error state and that the system is non-operational.**

Verdict:

PASS.

4.1.9.6 FPT_TST_EXT.3 SELF-TEST WITH DEFINED METHODS (MOD_VPNGW_V1.3)

4.1.9.6.1 FPT_TST_EXT.3 TSS (MOD_VPNGW_V1.3)

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

Evaluator Findings:
The evaluator reviewed the TSS, **section 6.8.2 labeled "FPT_FLS.1/SelfTest, FPT_TST_EXT.1, FPT_TST_EXT.3"**, to ensure that it describes the method used to perform self-testing on the TSF executable code, and that this method is consistent with what is described in the SFR. **The integrity of the firmware is tested using the cryptographic services described in FCS_COP.1/SigGen, which covers all of the executable code.**

This method is consistent with what is described in the SFR.

Verdict:
PASS.

4.1.9.6.2 FPT_TST_EXT.3 AGD (MOD_VPNGW_V1.3)

There are no guidance EAs for this component.

4.1.9.7 FPT_TUD_EXT.1 TRUSTED UPDATE (CPP_ND_V3.0E)

4.1.9.7.1 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 shall describe how and when the inactive version becomes active. The evaluator shall verify this description.

Evaluator Findings:
The evaluator verified that the TSS describes how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the TSS describes how and when the inactive version becomes active. The evaluator verified this description. The relevant information is found in the following section(s): **TOE Summary Specification section 6.8.4 labeled "FPT_TUD_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Authorized administrators can query the current version of the TOE software via the administration GUI.**

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 of the software before installation and that installation fails if the verification fails.

Evaluator Findings:
The evaluator verified 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 verified that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails. The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification section 6.8.4 labeled "FPT_TUD_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Updates are downloaded from a specified URI using HTTPS. Administrators can manually load the update file on the GoSilent Server. An option is provided for administrators to download GoSilent Cube updates to itself from the CACI update server. The updates are signed with a CACI key. Once the image has been downloaded, the TOE checks the signature of the image (against the CACI public key) before the image is applied.**

The evaluator shall verify that the TSS describes the method by which the digital signature is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature of the update, and the actions that take place for both successful and unsuccessful signature verification.

Evaluator Findings:
The evaluator verified that the TSS describes the method by which the digital signature is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature of the update, and the actions that take place for both successful and unsuccessful signature verification. The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification TOE Summary Specification section 6.8.4 labeled "FPT_TUD_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Updates are downloaded from a specified URI using HTTPS. Administrators can manually load the update file on the GoSilent Server. An option is provided for administrators to download GoSilent Cube updates to itself from the CACI update server. The updates are signed with a CACI key. Once the image has been downloaded, the TOE checks the signature of the image (against the CACI public key) before the image is applied.**

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.

Evaluator Findings:
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 verified that the TSS explains what actions are involved in automatic checking or automatic updating by the TOE, respectively. The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification section 6.8.4 labeled "FPT_TUD_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **The administrator can initiate a manual update of the TOE component that they are connected to.** **The TOE does not perform automatic updates.**

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

Evaluator Findings:
The evaluator examined the TSS and ensured that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component. Alternatively, this description can be provided in the guidance documentation. In that case the evaluator should examine the guidance documentation instead. The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification TOE Summary Specification section 6.8.4 labeled "FPT_TUD_EXT.1"** Upon investigation, the evaluator found that the TSS states that: **Administrators can manually load the update file on the GoSilent Server. An option is provided for administrators to download GoSilent Cube updates to itself from the CACI update server. The updates are signed with a CACI key. Once the image has been downloaded, the TOE checks the signature of the image (against the CACI public key) before the image is applied.**

Verdict:

PASS.

4.1.9.7.2 FPT_TUD_EXT.1 AGD

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.

Evaluator Findings:

The evaluator verified 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 describes how to query the loaded but inactive version.
The relevant information is found in the following section(s): **"Installation and Verification"**
 	
Upon investigation, the evaluator found that the AGD activity states that: **The TOE software comes preinstalled on the shipped hardware. The installed software version can be checked by doing the following:**
 	
• **GoSilent Server**
  ○ **Log into the GSS GUI as described in section 3.1 and navigate to the following:**
 	
  Maintenance > About
 	
• **GoSilent Cube**
  ○ **Log into the Cube GUI as described in section 3.1 and navigate to the following:**
 	
  Help > About This GoSilent

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

Evaluator Findings:
The evaluator verified that the guidance documentation describes how the verification of the authenticity of the update is performed (digital signature verification). The description includes the procedures for successful and unsuccessful verification. The description corresponds to the description in the TSS. The relevant information is found in the following section(s): **"Installation and Verification"** Upon investigation, the evaluator found that the AGD activity states that: **Authorized administrators can perform manual updates to the TOE software by connecting to the component in which they wish to update. Updates are downloaded from a specified URI using HTTPS. All update files are signed with a CACI security key. This signature is checked against the CACI Security Public Key by the TOE upon download and before installation. If the signature check or package upload fails, both the Cube and GSS will not proceed with the installation and instead continue executing the current version. CACI should be contacted for support if subsequent failures occur.**

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

Evaluator Findings:
The evaluator examined the guidance documentation and ensured that it describes how the versions of individual TOE components are determined for FPT_TUD_EXT.1, how all TOE components are updated, and the error conditions that may arise from checking or applying the update (e.g. failure of signature verification, or exceeding available storage space) along with appropriate recovery actions. The guidance documentation only has to describe the procedures relevant for the Security Administrator; it does not need to give information about the internal communication that takes place when applying updates. The relevant information is found in the following section(s): **"Installation and Verification"** Upon investigation, the evaluator found that the AGD activity states that: **The TOE software comes preinstalled on the shipped hardware. The installed software version can be checked by doing the following:** • **GoSilent Server** ○ **Log into the GSS GUI as described in section 3.1 and navigate to the following:** **Maintenance > About** • **GoSilent Cube** ○ **Log into the Cube GUI as described in section 3.1 and navigate to the following:** **Help > About This GoSilent** **Authorized administrators can perform manual updates to the TOE software by connecting to the component in which they wish to update. Updates are downloaded from a specified URI using HTTPS. All update files are signed with an ID Technologies security key. This signature is checked against the ID Technologies Security Public Key by the TOE upon download and before installation. If the signature check or package upload fails, both the Cube and GSS will not proceed with the installation and instead continue executing the current version. ID Technologies should be contacted for support if subsequent failures occur.**

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

Evaluator Findings:
If this was information was not provided in the TSS: For distributed TOEs, the evaluator examined the Guidance Documentation and ensured that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component. The relevant information is found in the following section(s): **"Installation and Verification"** Upon investigation, the evaluator found that the AGD activity states that: **Prior to the installation of a software update, the Cube will tear down any active tunnels before the update is applied to ensure the integrity of the update process. Once the update process has completed successfully on the Cube, the tunnel is reestablished. On GSS, when an update is successfully installed, GSS will reboot automatically, after which the update will take effect.** **Each GoSilent Cube must be registered on the GoSilent Server. Registration involves several steps to be completed by an administrator prior to a Cube establishing a connection to GSS. These steps can be completed by navigating to the 'VPN Clients' tab in GSS and clicking on the 'Activate' button next to a listed cube that has been licensed for GSS. The 'Add A Client' window will appear where the 'Device Serial Number' is listed but not configurable. Fill in the 'Client ID' and 'Client Description' fields and click 'Add Client'. Once these steps to prepare the Cube for registration have been completed, a connection attempt from the Cube to GSS can be made.** **Note: The first connection attempt from the Cube to GSS will complete the enablement process and may take longer to establish than subsequent connection attempts.** **The GoSilent Cube is always the initiator for connections to GSS, therefore should the connection be unintentionally broken during the registration process, the user or administrator should check all network connections for the GSS and the Cube to ensure the integrity of the network and then restart the enablement process described above. Should the enablement process not complete successfully, the administrator can remove the Cube in GSS by navigating to 'VPN Clients' and clicking the trash can icon ('Remove') next to the Cube experiencing issues, reboot the Cube device, and restarting the Enablement process again.** **Cubes may also be suspended and thereby subsequent connections from a cube can be rejected, by also navigating to the 'VPN Clients' tab in GSS and clicking on the pencil icon beside the Cube to be suspended, toggle the "Suspend" button, and click "Save".**

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 shall also ensure that the Guidance Documentation describes how the certificates are installed/updated/selected, if necessary.

Evaluator Findings:
N/A – The TOE does not support certificate-based mechanism for trusted updates.

Verdict:

PASS.

4.1.10 TOE ACCESS (FTA)

4.1.10.1 FTA_SSL_EXT.1 TSF-INITIATED SESSION LOCKING (CPP_ND_V3.0E)

4.1.10.1.1 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.

Evaluator Findings:
The evaluator examined the TSS to determine that it details whether local administrative session locking or termination is supported and the related inactivity time period settings. The relevant information is found in the following section(s): **TOE Summary Specification section 6.9.1 labeled "FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4"** Upon investigation, the evaluator found that the TSS states that: **Following an administrator configured period of inactivity (of both local and remote interactive sessions) the session will be automatically terminated, requiring re-authentication by the administrator before the access to TOE functionality can be gained.**

Verdict:

PASS.

4.1.10.1.2 FTA_SSL_EXT.1 AGD

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.

Evaluator Findings:
The evaluator confirmed that the guidance documentation states whether local administrative session locking or termination is supported and instructions for configuring the inactivity time period. The relevant information is found in the following section(s): **"Session Termination"** Upon investigation, the evaluator found that the AGD activity states that: **GSS supports automatic termination of inactive local and remote administrative sessions, in addition to manual session termination by an administrator. To modify these functions and the thresholds of GSS, login to the GSS GUI and navigate to the following:** **'System Settings > Admin Console'** **and scroll down to the following settings for administrator defined configuration:** • **'Session inactivity logout interval' (Note: this parameter must not be '0')**

Verdict:

PASS.

4.1.10.2 FTA_SSL.3 TSF-INITIATED TERMINATION (CPP_ND_V3.0E)

4.1.10.2.1 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.

Evaluator Findings:
The evaluator examined the TSS and determined that it details the administrative remote session termination and the related inactivity time period. The relevant information is found in the following section(s): **TOE Summary Specification section 6.9.1 labeled "FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4"** Upon investigation, the evaluator found that the TSS states that: **Following an administrator configured period of inactivity (of both local and remote interactive sessions) the session will be automatically terminated, requiring re-authentication by the administrator before the access to TOE functionality can be gained.**

Verdict:

PASS.

4.1.10.2.2 FTA_SSL.3 AGD

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

Evaluator Findings:
The evaluator confirmed that the guidance documentation includes instructions for configuring the inactivity time period for remote administrative session termination. The relevant information is found in the following section(s): **"Session Termination"** Upon investigation, the evaluator found that the AGD activity states that: **The Cube also supports automatic termination of inactive local sessions, in addition to manual session termination. To modify these functions and the thresholds of the Cube, log into the GSS GUI and navigate to the following:** **'System Settings > GoSilent'** **and scroll down to the following settings for administrator defined configuration:** • **'Session inactivity logout interval' (Note: this parameter must not be '0')** • **'Session inactivity warning interval' (Note: Must be less than the logout interval if the logout interval is not 0)**

Verdict:

PASS.

4.1.10.3 FTA_SSL.4 USER-INITIATED TERMINATION (CPP_ND_V3.0E)

4.1.10.3.1 FTA_SSL.4 TSS

The evaluator shall examine the TSS to determine that it details how the remote administrative session (and if applicable the local administrative session) are terminated.

Evaluator Findings:
The evaluator examined the TSS and determined that it details how the remote administrative session (and if applicable the local administrative session) are terminated. The relevant information is found in the following section(s): **TOE Summary Specification section 6.9.1 labeled "FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4"** Upon investigation, the evaluator found that the TSS states that: **The administrator is able to terminate their GUI session by using the logout button.**

Verdict:

PASS.

4.1.10.3.2 FTA_SSL.4 AGD

The evaluator shall confirm that the guidance documentation states how to terminate a remote interactive session (and if applicable the local administrative session).

Evaluator Findings:
The evaluator confirmed that the guidance documentation states how to terminate a remote interactive session (and if applicable the local administrative session). The relevant information is found in the following section(s): **"Session Termination"** Upon investigation, the evaluator found that the AGD activity states that: **A user may terminate their own session by navigating to the user icon in the top right corner of the GUI and selecting '*Logout*'.** **An administrator may terminate their own session by navigating to the user icon in the top right corner of the GUI and selecting '*Logout*'.**

Verdict:

PASS.

4.1.10.4 FTA_TAB.1 DEFAULT TOE ACCESS BANNERS (CPP_ND_V3.0E)

4.1.10.4.1 FTA_TAB.1 TSS

The evaluator shall check the TSS to ensure that it details each administrative method of access (local and/or remote) available to the Security Administrator (e.g., serial port, SSH, HTTPS).

Evaluator Findings:
The evaluator checked the TSS and ensured that it details each administrative method of access (local and remote) available to the Security Administrator (e.g., serial port, SSH, HTTPS). The relevant information is found in the following section(s): **TOE Summary Specification section 6.9.2 labeled "FTA_TAB.1"** Upon investigation, the evaluator found that the TSS states that: **The TOE displays an administrator configurable message prior to local and remote login via the GoSilent Server GUI. The Cube also displays an administrator configurable message prior to local login via the Cube GUI.**

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).

Evaluator Findings:
The evaluator checked the TSS and ensured 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). The relevant information is found in the following section(s): **TOE Summary Specification section 6.9.2 labeled "FTA_TAB.1"** Upon investigation, the evaluator found that the TSS states that: **Prior to local and remote login to GoSilent Server and local login to the Cube, the GUI displays a consent banner to the administrator warning that proceeding with authentication is consenting to the terms of use of the TOE. The user is then prompted to enter their username and password.**

Verdict:

PASS.

4.1.10.4.2 FTA_TAB.1 AGD

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

Evaluator Findings:
The evaluator examined the guidance documentation and ensured that it describes how to configure the banner message. The relevant information is found in the following section(s): **"Login Banner"** Upon investigation, the evaluator found that the AGD activity states that: **To modify the login banner text for GSS, log into the GSS GUI and navigate to the following:** ***'System Settings > Admin Console'*** **and scroll down to the 'Enable Login Banner' setting and ensure the box is checked.** **Continue scrolling down to the 'Login Banner Text' setting and enter the desired login banner message to be displayed.** **To modify the login banner text for Cube, log into the GSS GUI and navigate to the following:** ***'System Settings > GoSilent'*** **and scroll down to the 'Enable Login Banner' setting and ensure the box is checked.** **Continue scrolling down to the 'Login Banner Text' setting and enter the desired login banner message to be displayed.**

Verdict:

PASS.

4.1.11 TRUSTED PATH/CHANNELS (FTP)

4.1.11.1 FTP_ITC.1 INTER-TSF TRUSTED CHANNEL (CPP_ND_V3.0E)

4.1.11.1.1 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.

Evaluator Findings:
The evaluator examined the TSS and determined 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 relevant information is found in the following section(s): **TOE Summary Specification section 6.10.1 labeled "FTPITC.1, FTP_ITC.1/VPN"**

Upon investigation, the evaluator found that the TSS states that: **Trusted channels are used for connections between GoSilent Cube and GoSilent Server as well as with the syslog server. TLS syslog server connections are always initiated by GoSilent Server and GoSilent Cube.**

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.

Evaluator Findings:
The evaluator also confirmed 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. The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification section 6.10.1 labeled "FTPITC.1, FTP_ITC.1/VPN"** Upon investigation, the evaluator found that the TSS states that: **Trusted channels are used for connections between GoSilent Cube and GoSilent Server (FCS_IPSEC_EXT.1) as well as with the syslog server (FCS_TLSC_EXT.1). TLS syslog server connections are always initiated by GoSilent Server/GoSilent Cube (acting as the TLS client); assured identification is via validation of the server X.509 certificate. IPsec connections between TOE components are always initiated by the GoSilent Cube.**

Verdict:

PASS.

4.1.11.1.2 FTP_ITC.1 AGD

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.

Evaluator Findings:
The evaluator confirmed that the AGD contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken. The relevant information is found in the following section(s): **"Audit Logging"** Upon investigation, the evaluator found that the AGD activity states that: **Each TOE component also transmits a copy of each audit record to an external syslog server in real time via TLS. Syslog server parameters are configured on the GoSilent Server and communicated to all GoSilent Cubes.**

Verdict:

PASS.

4.1.11.2 FPT_ITC.1/VPN

4.1.11.3 FTP_TRP.1/ADMIN TRUSTED PATH (CPP_ND_V3.0E)

4.1.11.3.1 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.

Evaluator Findings:
The evaluator examined the TSS and determined that the methods of remote TOE administration are indicated, along with how those communications are protected. The relevant information is found in the following section(s): **TOE Summary Specification section 6.10.2 labeled "FTP_TRP.1/Admin"** Upon investigation, the evaluator found that the TSS states that: **The TOE provides a trusted path for remote administration of the GoSilent Server and GoSilent Cube. These logically distinct communications paths use TLS and HTTP to protect any transmitted data from unauthorized disclosure and leverage the ciphersuites listed in FCS_TLSS_EXT.1.**

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.

Evaluator Findings:
The evaluator also confirmed 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. The relevant information is found in the following section(s): **TOE Summary Specification TOE Summary Specification section 6.10.2 labeled "FTP_TRP.1/Admin"** Upon investigation, the evaluator found that the TSS states that: **The TOE provides a trusted path for remote administration of the GoSilent Server and GoSilent Cube. These logically distinct communications paths use TLS and HTTP to protect any transmitted data from unauthorized disclosure and leverage the ciphersuites listed in FCS_TLSS_EXT.1.**

Verdict:

PASS.

4.1.11.3.2 FTP_TRP.1/ADMIN AGD

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

Evaluator Findings:
The evaluator confirmed that the guidance documentation contains instructions for establishing the remote administrative sessions for each supported method.

The relevant information is found in the following section(s): **"Administration Interfaces"**

Upon investigation, the evaluator found that the AGD activity states that: **Only the following administration interfaces may be used:**

a) GoSilent Server GUI. Administrative GUI interface for managing all GoSilent configurations on the server and other GoSilent platforms.

o Local access is provided for the configured whitelisted source IP address of an administrator device, and by navigating to the local IP address of GSS through the (virtual) MGMT Ethernet interface using HTTPS via web browser to TCP port 4439.

o Remote access is provided to administrators connecting from Cube clients (for any client that has a valid GSS administrator account) using HTTPS via web browser to the MGMT IP address of the GSS over TCP port 443.

b) GoSilent Cube GUI. User GUI interface that provides limited functionality for initial Cube configuration.

o Local access is provided to users by navigating to the local IP address of the cube using HTTPS via web browser over port 4439 or by navigating to the URL https://setup.gosilent.

o Configuration of some Cube functions is done via the GSS GUI. The Cube subsequently downloads these configuration settings from GSS upon connection establishment.

Verdict:

PASS.

5 SECURITY ASSURANCE REQUIREMENTS

5.1 ASE: SECURITY TARGET EVALUATION

5.1.1 GENERAL EVALUATION ACTIVITIES FOR TOE SUMMARY SPECIFICATION (ASE_TSS.1) FOR ALL TOES (CPP_ND_V3.0E)

5.1.1.1 EVALUATION ACTIVITY

When evaluating a Security Target, the evaluator performs the work units as presented in the CEM. In addition, the evaluator shall ensure the content of the TSS in the ST satisfies the EAs specified in Section 2 (Evaluation Activities for SFRs).

Evaluator Findings:
The evaluator performed the work units as presented in the CEM. Furthermore, the evaluator examined the TTS and ensured the content of the TSS in the ST satisfies the EAs specified in Section 2 (Evaluation Activities for SFRs).

Verdict:

PASS.

5.2 ADV: DEVELOPMENT

5.2.1 BASIC FUNCTIONAL SPECIFICATION (ADV_FSP.1) (CPP_ND_V3.0E)

5.2.1.1 EVALUATION ACTIVITY

The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g. audit review or performing updates). Explicitly labeling TSFI as security relevant or non-security relevant is not necessary. A TSFI is implicitly security relevant if it is used to satisfy an evaluation activity, or if it is identified in the ST or guidance documentation as adhering to the security policies (as presented in the SFRs). The intent is that these interfaces will be adequately tested and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied. According to the description above 'security relevant' corresponds to the combination of 'SFRenforcing' and 'SFR-supporting' as defined in CC Part 3, paragraph 224 and 225.

The set of TSFI that are provided as evaluation evidence are contained in the Security Target and the guidance documentation.

Evaluator Findings:
TOE Design information that can be made public is available in the guidance documentation and in the ST. Any sensitive or proprietory information required by this protection profile is not to be made public. It is not necessary to proivde a complete specification of the TSFIs. For NDcPP, additional “functional specification” documentation is not necessary because this requirement is satisfied by multiple other documents (AGD, TSS, and Testing). All associated activities are covered in the Test Report, ST, and AGD documents. NDcPP3.0e, section 7.2.1 states that: “For this cPP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation.” All of the above information is applicable to the ADV Evaluation Activities (5.2.1.1, 5.2.1.2, and 5.2.1.3) in NDcPP3.0e-SD. The evaluator examined the ST (Security Target) and the AGD (interface documentation) to verify that it describes the purpose and method of use for each TSFI that is identified as being security relevant. The evaluator examined the entire AGD. The evaluator verified the AGD describes the purpose and method of use for each security relevant TSFI by verifying the AGD satisfies all the AGD Evaluation Activities. During testing, the evaluator used the product and its interfaces extensively and did not find any areas that were deficient.

Verdict:
PASS.

5.2.1.2 EVALUATION ACTIVITY

The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant.

Evaluator Findings:
The evaluator checked the interface documentation (AGD) and ensured it identifies and describes the parameters for each TSFI that is identified as being security relevant. This is covered in the previous evaluation activity above.

Verdict:
PASS.

5.2.1.3 EVALUATION ACTIVITY

The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

The evaluator shall use the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in Section 2, including the EAs associated with testing of the interfaces.

It should be noted that there may be some SFRs that do not have a TSFI that is explicitly "mapped" to invoke the desired functionality. For example, generating a random bit string or destroying a cryptographic key that is no longer needed are capabilities that may be specified in SFRs, but are not invoked by an interface.

The required EAs define the design and interface information required to meet ADV_FSP.1. If the evaluator is unable to perform some EA, then the evaluator shall conclude that an adequate functional specification has not been provided.

Evaluator Findings:
The evaluator examined the interface documentation to develop a mapping of the interfaces to SFRs. The evaluator used the provided documentation to first identify, and then examine a representative set of interfaces to perform the EAs presented in Section 2, including the EAs associated with testing of the interfaces. This is covered in the previous evaluation activity above.

Verdict:
PASS.

5.3 AGD: GUIDANCE DOCUMENTATION

5.3.1 OPERATIONAL USER GUIDANCE (AGD_OPE.1) (CPP_ND_V3.0E)

5.3.1.1 EVALUATION ACTIVITY

The evaluator shall ensure 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.

Evaluator Findings:
The evaluator checked the requirements above are met by the AGD. The AGD is distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration. Upon investigation, the evaluator found that the CC guidance will be published with the CC certificate on www.niap-ccevs.org.

Verdict:
PASS.

5.3.1.2 EVALUATION ACTIVITY

The evaluator shall ensure that the Operational guidance is provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

Evaluator Findings:

The evaluator ensured that the AGD is provided for every Operational Environment that the product supports as claimed in the Security Target. The section titled **"Evaluated Software and Hardware"** of the AGD was used to determine the verdict of this assurance activity. The AGD specifies that the platforms supported are:
 	
The Target of Evaluation (TOE) is the CACI idt GoSilent Server and Cube v26.01:
 	
a) Build: 26.01.x (GoSilent Server)
b) Build: 26.01.x (GoSilent Cube)
 	
The operational environments for the TOE are:
 	
• **GoSilent Server**
  ○ **Debian 12 on Red Hat RHEL v9.6 KVM on Intel Xeon Silver 4514Y (Raptor Cove)**
  ○ **Debian 12 on VMware ESXi v8.0 on Intel Xeon Silver 4514Y (Raptor Cove)**
• **Cube**
  ○ **Armbian 25.05 on AllWinner H5/ Cortex-A53 (ARM 8-A)**
 	
The Cube is delivered via commercial courier with the software pre-installed. GoSilent Server is a software-only distribution.

Verdict:
PASS.

5.3.1.3 EVALUATION ACTIVITY

The evaluator shall ensure that the Operational guidance contains instructions for configuring any cryptographic implementation associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic implementations was not evaluated nor tested during the CC evaluation of the TOE.

Evaluator Findings:
The evaluator ensured that the AGD section **"Cryptographic Functions"** contains instructions for configuring any cryptographic implementation associated with the evaluated configuration of the TOE. It provides a warning to the administrator that use of other cryptographic implementations was not evaluated nor tested during the CC evaluation of the TOE.

Verdict:
PASS.

5.3.1.4 EVALUATION ACTIVITY

The evaluator shall ensure the Operational guidance makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

Evaluator Findings:
The entire AGD was used to determine the verdict of this work unit. Each confirmation command indicates tested options. Additionally, the section titled **"Functions Not Included in the TOE Evaluation"** specifies features that are not assessed and tested by the EAs. The evaluator ensured the AGD makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

Verdict:
PASS.

5.3.1.5 EVALUATION ACTIVITY

In addition, the evaluator shall ensure that the following requirements are also met:

a. The guidance documentation shall contain instructions for configuring any cryptographic implementation associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic implementations was not evaluated nor tested during the CC evaluation of the TOE.

b. The evaluator shall verify that this process includes instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).

c. The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The guidance documentation shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.

Evaluator Findings:
The evaluator verified the AGD contains instructions for configuring any cryptographic implementations in **"Cryptographic Functions"** The evaluator verified the AGD describes the process for verifying updates in **"Installation and Verification"** The evaluator verified the AGD makes it clear which security functionality is covered by the Evaluation Activities in **"Conformance Claims"**

Verdict:
PASS.

5.3.2 PREPARATIVE PROCEDURES (AGD_PRE.1) (CPP_ND_V3.0E)

5.3.2.1 EVALUATION ACTIVITY

The evaluator shall examine the Preparative procedures to ensure 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).

The documentation should be in an informal style and should be written with sufficient detail and explanation that they can be understood and used by the target audience (which will typically include IT staff who have general IT experience but not necessarily experience with the TOE product itself).

Evaluator Findings:

The evaluator examined the Preparative procedures to ensure they include a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality. The evaluator reviewed the sections titled **"Evaluated Software and Hardware"** of the AGD. The evaluator found that these sections describe how the Operational Environment must meet:
 	
The operational environments for the TOE are:
 	
- GoSilent Server
  ○ Debian 12 on Red Hat RHEL v9.6 KVM on Intel Xeon Silver 4514Y (Raptor Cove)
  ○ Debian 12 on VMware ESXi v8.0 on Intel Xeon Silver 4514Y (Raptor Cove)
- Cube
  ○ Armbian 25.05 on AllWinner H5/ Cortex-A53 (ARM 8-A)

Verdict:
PASS.

5.3.2.2 EVALUATION ACTIVITY

The evaluator shall examine the preparative procedures to ensure they are provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

Evaluator Findings:

The evaluator checked the requirements above are met by the preparative procedures. The entire AGD was used to determine the verdict of this work unit. Upon investigation, the evaluator found that the AGD describes each of the devices in the operating environment.
 	
The operational environments for the TOE are:
 	
- GoSilent Server
  ○ Debian 12 on Red Hat RHEL v9.6 KVM on Intel Xeon Silver 4514Y (Raptor Cove)
  ○ Debian 12 on VMware ESXi v8.0 on Intel Xeon Silver 4514Y (Raptor Cove)
- Cube
  ○ Armbian 25.05 on AllWinner H5/ Cortex-A53 (ARM 8-A)

Verdict:
PASS.

5.3.2.3 EVALUATION ACTIVITY

The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment.

Evaluator Findings:
The evaluator checked the requirements are met by the preparative procedures. The entire AGD was used to determine the verdict of this work unit. Upon investigation, the evaluator found that AGD describes all of the instructions necessary to install and configure the TOE to work in the target operating environment, including: - Configuring Administrative Accounts and Passwords - Configuring the Remote Syslog Server - Configuring Audit Log Options - Configuring Event Logging - Configuring a Secure Logging Channel - Configuring VPNs (IPsec) - Configuring Security Flow Policies - Configuring Traffic Filtering Rules

Verdict:
PASS.

5.3.2.4 EVALUATION ACTIVITY

The evaluator shall examine the preparative procedures to ensure they include instructions on how to manage the TSF as a product and as a component of the larger Operational Environment in a manner that allows to preserve integrity of the TSF.

The intent of this requirement is to ensure there exists adequate preparative procedures (guidance in most cases) to put the TSF in a secure state (i.e., evaluated configuration). AGD_PRE.1 lists general requirements, the specific assurance activities implementing it are performed as part of FMT_SMF.1, FMT_MTD.1 and FMT_MOF.1 series of SFRs.

Evaluator Findings:
The evaluator ensured the preparative procedures include instructions to manage the TSF as a product and as a component of the larger operational environment in a manner that preserves the integrity of the TSF. The entire AGD was used to determine the verdict of this work unit. The same commands, configurations, and interfaces used to install the TOE are also used for ongoing management, so this is satisfied by AGD_PRE.1 Test #3.

Verdict:
PASS.

5.3.2.5 EVALUATION ACTIVITY

In addition, the evaluator shall ensure that the following requirements are also met. The preparative procedures must:

- include instructions to provide a protected administrative capability; and
- identify TOE passwords that have default values associated with them and mandate that they shall be changed.

Evaluator Findings:
The evaluator ensured the preparative procedures include instructions to provide a protected administrative capability and changing default passwords. The sections titled **"Administration Interfaces"** were used to determine the verdict of this work unit. The AGD describes: Only the following administration interfaces may be used: i) **GoSilent Server GUI.** Administrative GUI interface for managing all GoSilent configurations on the server and other GoSilent platforms. ○ Local access is provided for the configured whitelisted source IP address of an administrator device, and by navigating to the local IP address of GSS through the (virtual) MGMT Ethernet interface using HTTPS via web browser to TCP port 4439. ○ Remote access is provided to administrators connecting from Cube clients (for any client that has a valid GSS administrator account) using HTTPS via web browser to the MGMT IP address of the GSS over TCP port 443. j) **GoSilent Cube GUI.** User GUI interface that provides limited functionality for initial Cube configuration. ○ Local access is provided to users by navigating to the local IP address of the cube using HTTPS via web browser over port 4439 or by navigating to the URL https://setup.gosilent. ○ Configuration of some Cube functions is done via the GSS GUI. The Cube subsequently downloads these configuration settings from GSS upon connection establishment.

Verdict:
PASS.

5.4 AVA: VULNERABILITY ASSESSMENT

5.4.1 VULNERABILITY SURVEY (AVA_VAN.1) (CPP_ND_V3.0E)

5.4.1.1 EVALUATION ACTIVITY (DOCUMENTATION)

In addition to the activities specified by the CEM in accordance with Table 2, the evaluator shall perform the following activities.

The evaluator shall examine the documentation outlined below provided by the developer to confirm that it contains all required information. This documentation is in addition to the documentation already required to be supplied in response to the EAs listed previously.

The developer shall provide documentation identifying the list of software and hardware components that compose the TOE. Hardware components should identify compute-capable hardware components, at a minimum that must include the processor, and where applicable, discrete crypto ASICs, TPMs, etc. 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 implementations, (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.

If the TOE is a distributed TOE then the developer shall provide:

a. documentation describing the allocation of requirements between distributed TOE components as in [NDcPP, 3.4]

b. a mapping of the auditable events recorded by each distributed TOE component as in [NDcPP, Table 2]

c. additional information in the Preparative Procedures as identified in the refinement of AGD_PRE.1 in additional information in the Preparative Procedures as identified in 3.4.1.2 and 3.5.1.2.

Evaluator Findings:
The evaluator collected this information from the developer which was used to feed into the Public Domain Search. Refer to evaluator findings in the evaluation activity below.

Verdict:
PASS.

5.4.1.2 EVALUATION ACTIVITY

The evaluator shall formulate hypotheses in accordance with process defined in Appendix A. The evaluator shall document 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.

Evaluator Findings:
The evaluator documented their analysis and testing of potential vulnerabilities with respect to this requirement. Public searches were performed against all keywords found within the Security Target and AGD that may be applicable to specific TOE components. This included protocols, TOE software version, and TOE hardware to ensure sufficient coverage under AVA. The evaluator searched the Internet for potential vulnerabilities in the TOE using the web sites listed below. The sources of the publicly available information are provided below: • https://nvd.nist.gov/view/vuln.search • http://cve.mitre.org/cve The evaluator examined public domain vulnerability searches by performing a keyword search. The terms used for this search were based on the vendor name, product name, and key platform features leveraged by the product. As a result, the evaluator performed a search using the following keywords: • (Openssl 3.0.17) cpe:/:openssl:openssl:3.0.17 • (dhcpd 9.4.1) cpe:/:isc:dhcp:9.4.1 • (nginx 1.22.1) cpe:/:nginx:nginx:1.22.1 • (noVNC 1.3.0) cpe:/:novnc:novnc:1.3.0 • (python3 3.9.6) cpe:/:python:python:3.9.6 • (strongswan 6.0.0) cpe:/:strongswan:strongswan:6.0.0 • (chronyd 4.3) cpe:/:chrony_project:chrony:4.3 • (syslog-ng 3.38.1) cpe:/:balabit:syslog-ng:3.38.1 • (APScheduler 3.5.1) cpe:/:apscheduler:apscheduler:3.5.1 • (asciimatics 1.11.0) cpe:/:asciimatics_project:asciimatics:1.11.0 • (Authlib 0.15.4) cpe:/:authlib:authlib:0.15.4 • (Babel 2.5.3) cpe:/:babel_project:babel:2.5.3 • (beautifulsoup4 4.10.0) cpe:/:beautifulsoup4_project:beautifulsoup4:4.10.0 • (blinker 1.4) cpe:/:blinker_project:blinker:1.4

- (certify 2018.4.16) cpe:/:certify_project:certify:2018.4.16
- (chardet 3.0.4) cpe:/:chardet_project:chardet:3.0.4
- (ciso8601 2.0.1) cpe:/:ciso8601_project:ciso8601:2.0.1
- (click 6.7) cpe:/:pallets:click:6.7
- (configparser 4.0.2) cpe:/:configparser_project:configparser:4.0.2
- (coverage 4.5.1) cpe:/:coverage_project:coverage:4.5.1
- (cryptography 3.1.1) cpe:/:cryptography_project:cryptography:3.1.1
- (dnspython 1.16.0) cpe:/:dnspython:dnspython:1.16.0
- (entrypoints 0.3) cpe:/:entrypoints_project:entrypoints:0.3
- (enum34 1.1.6) cpe:/:enum34_project:enum34:1.1.6
- (Flask 1.0.2) cpe:/:pallets:flask:1.0.2
- (flake8 3.6.0) cpe:/:flake8_project:flake8:3.6.0
- (netaddr 0.7.19) cpe:/:netaddr:netaddr:0.7.19
- (passlib 1.7.1) cpe:/:passlib:passlib:1.7.1
- (Pillow 9.5.0) cpe:/:python-pillow:pillow:9.5.0
- (python-iptables 1.0.0) cpe:/:python-iptables_project:python-iptables:1.0.0
- (SQLAlchemy 1.2.7) cpe:/:sqlalchemy:sqlalchemy:1.2.7
- (urllib3 1.23) cpe:/:urllib3:urllib3:1.23
- (Werkzeug 0.14.1) cpe:/:pallets:werkzeug:0.14.1
- (WTForms 2.1) cpe:/:wtforms_project:wtforms:2.1
- (bzip 1.0.8-5+b1) cpe:/a:bzip:bzip2:1.0.8
- CACI
- CACI GoSilent
- CACI Cube

The initial searches were performed on 8/29/2025, 03/06/2026 and 05/12/2026 and a final search was performed on 06/24/2026. No open vulnerabilities applicable to the TOE were identified.

The evaluation lab examined each result provided from NVD and Exploit Search to determine if the current TOE version or component within the environment was vulnerable. Based upon the analysis, any issues found that were generated were patched in the TOE version and prior versions, mitigating the risk factor.

Verdict:
PASS.

6.1.1 SECURITY AUDIT (FAU)

6.1.1.1 FAU_GEN.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 identity and authentication (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. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.
Test Steps	• The evaluator triggers each auditable event on the TOE. • The evaluator verifies that each audit record is generated and contains the required information. • The evaluator verifies all the audit record matches the audit records in all the test cases.
Pass/Fail with Explanation	Pass. The audit records associated with each test case are recorded with each test case. A comparison of required audit records to the presented audit records was additionally performed. This analysis shows that each required audit record is generated by the TOE.

6.1.1.2 FAU_GEN.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. The testing evidence in FAU_GEN.1 Test #1 is contained in each individual TOE component test report.

6.1.1.3 FAU_GEN.1/VPN TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	The evaluator shall test the audit functionality by performing actions that trigger each of the claimed audit events and verifying that the audit records are accurate and that their format is consistent with what is specified in the operational guidance. The evaluator may generate these audit events as a consequence of performing other tests that would cause these events to be generated.
Test Steps	• Trigger each auditable event on the TOE. Verify that each audit record is generated and contains the required information.
Pass/Fail with Explanation	Pass. The audit records associated with each test case are recorded with each test case. A comparison of required audit records to the presented audit records was additionally performed. This analysis shows that each required audit record is generated by the TOE.

6.1.1.4 FAU_GEN.1 TEST #1 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	In addition to the Evaluation Activities specified in the Supporting Document for the Base-PP, the evaluator shall perform tests to demonstrate that audit records are generated for the auditable events as specified in Table 2 of the PP-Module and, if the optional SFR FFW_RUL_EXT.2 is claimed by the TOE, Table 3.
Pass/Fail with Explanation	Pass. This test has been completed as part of FAU_GEN.1 Test #1 in the NDcPPv3.0e TR.

6.1.1.5 FAU_GEN.2 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.
Pass/Fail with Explanation	Pass. This activity has been exercised in FAU_GEN.1.

6.1.1.6 FAU_GEN.2 TEST #2 (CPP_ND_V3.0E)

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

6.1.1.7 FAU_GEN_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	For distributed TOEs, the requirements on TSS, Guidance Documentation and Tests regarding FAU_GEN_EXT.1 are already covered by the corresponding requirements for FAU_GEN.1.
Pass/Fail with Explanation	Pass. All requirements are met and already covered by the corresponding requirements in FAU_GEN.1.

6.1.1.8 FAU_STG_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: 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	• Record the audit server name and version information. • Configure a syslog server to communicate with the TOE. • Establish a TLS session between the remote syslog server and the TOE. • Verify that the logs generated on the TOE and the ones seen on the remote syslog server are the same. • Verify that the traffic between the TOE and Syslog is not sent in plaintext using packet capture.
Pass/Fail with Explanation	Pass. The TOE is capable of transferring audit data to an external audit server automatically without administrative intervention and it is protected by SSH when it is exported. This meets the testing requirement.

6.1.1.9 FAU_STG_EXT.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 2: For distributed TOEs, Test 1 defined above shall be applicable to all TOE components that forward audit data to an external audit server.
Pass/Fail with Explanation	Pass. All TOE components were tested individually in separate reports under FAU_STG_EXT.1 Test #1.

6.1.1.10 FAU_STG_EXT.1 TEST #3 (A) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 3: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall then make note of whether the TSS claims persistent or non-persistent logging and perform one of the following actions:

	i. If **persistent logging** is selected, the evaluator shall perform a power cycle of the TOE and ensure that following power on operations the log events generated are still maintained within the local audit storage.
Test Steps	• Perform an activity to generate audit data. • Verify that the above activity generates an audit log and note the same. • Restart the TOE and verify that the highlighted log in the above steps is still available in TOE's log store.
Pass/Fail with Explanation	Pass. The TOE retains old log events in its local audit storage even after it is rebooted. This meets the testing requirement.

6.1.1.11 FAU_STG_EXT.1 TEST #3 (B) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 3: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall then make note of whether the TSS claims persistent or non-persistent logging and perform one of the following actions: ii. If **non-persistent logging** is selected, the evaluator shall perform a power cycle of the TOE and ensure that following power on operations the log events generated are no longer present within the local audit storage.
Pass/Fail with Explanation	NA. Non-persistent logging is not selected in the ST, hence this test is not applicable.

6.1.1.12 FAU_STG_EXT.1 TEST #4 (A) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 4: 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.1.5. Depending on the configuration this means that the evaluator shall check the content of the audit data when the audit data is just filled to the maximum and then verifies that:

	i. 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.1.5).
Pass/Fail with Explanation	NA. Drop new audit data is not selected in FAU_STG_EXT.1.5, hence this test is not applicable.

6.1.1.13 FAU_STG_EXT.1 TEST #4 (B) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 4: 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.1.5. Depending on the configuration this means that the evaluator shall check the content of the audit data when the audit data is just filled to the maximum and then verifies that: ii. 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.1.5)
Test Steps	• Configure a local logfile on the TOE along with the overwriting behavior. • Perform log-generating actions. • Verify that the logs are locally stored and when the audit data is filled to the max, the existing audit data is overwritten.
Pass/Fail with Explanation	Pass. The TOE overwrites the oldest audit records when the local audit space is filled. This meets the testing requirement.

6.1.1.14 FAU_STG_EXT.1 TEST #4 (C) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 4: 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.1.5. Depending on the configuration this means

	that the evaluator shall check the content of the audit data when the audit data is just filled to the maximum and then verifies that: iii. The TOE behaves as specified (for the option '**other action'** in FAU_STG_EXT.1.5).
Pass/Fail with Explanation	NA. Other action is not selected in FAU_STG_EXT.1.5, hence this test is not applicable.

6.1.1.15 FAU_STG_EXT.1 TEST #5 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 5: For distributed TOEs, for the local storage according to FAU_STG_EXT.1.4, Test 1 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, Test 2 specified above shall be applied. The evaluator shall verify that the transfer of audit data to an external audit server is implemented.
Pass/Fail with Explanation	NA. All of the components of the TOE use local audit storage, but the TOE does not comply with FAU_STG_EXT.2, hence this test is not applicable.

6.1.1.16 FAU_STG_EXT.1 TEST #6 (CPP_ND_V3.0E) **[TD0886]**

Item	Data
Test Assurance Activity	Testing of secure transmission of the audit data externally (FTP_ITC.1) and, where applicable, intercomponent (FPT_ITT.1 or FTP_ITC.1) shall be performed according to the assurance activities for the particular protocol(s). The evaluator shall perform the following additional test for this requirement: Test 6 [Conditional]: In case manual export or ability to view locally is selected in FAU_STG_EXT.1.6, during interruption the evaluator shall perform a TSF-mediated action and verify the event is recorded in the audit trail. Note: The intent of the test is to ensure that the local audit TSF (as specified by FAU_STG_EXT.1.3) operates independently from the ability to transmit the generated audit data to an external audit server (as specified in FAU_STG_EXT.1.1). There are no specific requirements on the interruption of the connection between the TOE and the external audit server (as for FTP_ITC.1).

	TD0886 has been applied.
Test Steps	• Establish a TLS session between the remote syslog server and the TOE. • Verify that the TLS session was established using TOE logs. • Verify that the logs generated on the TOE for the above activity and those seen on the remote syslog server are the same. • Interrupt the connection between the remote syslog server and the TOE by shutting down the syslog server interface. • Attempt a config change on the device and verify logs are generated on the TOE. • Import the local TOE logfile to a remote entity. • Verify the logs after the interruption are imported on to the remote device.
Pass/Fail with Explanation	Pass. Logs generated using a TSF-mediated action are recorded in the audit trail during an interruption of the connection with the remote syslog server. This meets the testing requirement.

6.1.1.17 FAU_STG_EXT.2 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall verify that the numbers provided by the TOE according to the selection for FAU_STG_EXT.2 are correct when performing the tests for FAU_STG_EXT.1.5. For distributed TOEs the evaluator shall verify the correct implementation of counting of lost audit data for all TOE components that are supporting this feature according to the description in the TSS.
Pass/Fail with Explanation	NA. FAU_STG_EXT.2 is not selected in the ST, hence this test is not applicable.

6.1.1.18 FAU_STG_EXT.3 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall verify that a warning is issued by the TOE before the local storage space for audit data is full. For distributed TOEs the evaluator shall verify the correct implementation of display warning for local storage space for all TOE components that are supporting this feature according to the description in the TSS. The evaluator shall verify that each component that supports this feature according to the description in the TSS is capable of generating a warning itself or through another component.

Pass/Fail with Explanation	NA. FAU_STG_EXT.3 is not selected in the ST, hence this test is not applicable.

6.1.1.19 FAU_STG_EXT.4 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	For at least one of each type of distributed TOE components (sensors, central nodes, etc.), the following tests shall be performed using distributed TOEs. Test 1: For each type of TOE component, the evaluator shall perform a representative subset of auditable actions and ensure that these actions cause the generation of appropriately formed audit records. Generation of such records can be observed directly on the distributed TOE component (if there is appropriate interface), or indirectly after transmission to a central location. While performing these tests, the evaluator shall verify that the TOE behaviour observed during testing is consistent with the descriptions provided in the TSS and the Guidance Documentation. Depending on the TOE configuration, there might be a large number of different possible configurations. In such cases, it is acceptable to perform subset testing, accompanied by an equivalency argument describing the evaluator's sampling methodology.
Pass/Fail with Explanation	Pass. This test has been exercised in FAU_GEN.1 Test #1 in each respective test report. All audit logs are appropriately generated and stored on each individual TOE component.

6.1.1.20 FAU_STG_EXT.4 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	For at least one of each type of distributed TOE components (sensors, central nodes, etc.), the following tests shall be performed using distributed TOEs. Test 2: For each type of TOE component that, in the evaluated configuration, is capable of transmitting audit information to the external audit server (as specified in FTP_ITC.1), the evaluator shall configure a trusted channel and confirm that audit records generated as a result of actions taken by the evaluator are securely transmitted. It is sufficient to observe negotiation and establishment of the secure channel with the TOE component and the subsequent transmission of encrypted data to confirm this functionality. Alternatively, the following steps shall be performed: The evaluator shall induce audit record transmission, then review the

	packet capture around the time of transmission and verifies that no audit data is transmitted in the clear. While performing these tests, the evaluator shall verify that the TOE behaviour observed during testing is consistent with the descriptions provided in the TSS and the Guidance Documentation. Depending on the TOE configuration, there might be a large number of different possible configurations. In such cases, it is acceptable to perform subset testing, accompanied by an equivalency argument describing the evaluator's sampling methodology.
Pass/Fail with Explanation	Pass. This test has been exercised in FAU_GEN.1 Test #1 in each respective test report. All audit logs are appropriately generated and stored on each individual TOE component. All TOE components are also capable of transmitting audit information to the external audit server using the trusted channel found in FTP_ITC.1.

6.1.1.21 FAU_STG_EXT.4 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	For at least one of each type of distributed TOE components (sensors, central nodes, etc.), the following tests shall be performed using distributed TOEs. Test 3: For each type of TOE component that, in the evaluated configuration, is capable of transmitting audit information to another TOE component (as specified in FTP_ITT.1 or FTP_ITC.1, respectively), the evaluator shall configure a secure channel and confirm that audit records generated as a result of actions taken by the evaluator are securely transmitted. It is sufficient to observe negotiation and establishment of the secure channel with the TOE component and the subsequent transmission of encrypted data to confirm this functionality. Alternatively, the following steps shall be performed: The evaluator shall induce audit record transmission, then review the packet capture around the time of transmission and verifies that no audit data is transmitted in the clear. While performing these tests, the evaluator shall verify that the TOE behaviour observed during testing is consistent with the descriptions provided in the TSS and the Guidance Documentation. Depending on the TOE configuration, there might be a large number of different possible configurations. In such cases, it is acceptable to perform subset testing, accompanied by an equivalency argument describing the evaluator's sampling methodology.
Pass/Fail with Explanation	NA. Audit data is stored locally on each component of the TOE, hence this test is not applicable.

6.1.2 COMMUNICATION (FCO)

6.1.2.1 FCO_CPC_EXT.1 TEST #1A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	(Note: Section 3.5.1 lists questions for which the evaluator shall determine and report answers through the combination of the TSS, Guidance Documentation, and Tests Evaluation Activities.) The evaluator shall perform the following tests: Test 1a: the evaluator shall confirm that an IT entity that is not currently a member of the distributed TOE cannot communicate with any component of the TOE until the non-member entity is enabled by a Security Administrator for each of the non-equivalent TOE components[6] that it is required to communicate with (non-equivalent TOE components are as defined in the minimum configuration for the distributed TOE) The evaluator shall repeat Tests 1a and 1b for each different type of enablement process that can be used in the TOE.
Test Steps	• The evaluator attempts to connect to the GoSilent Server and GoSilent Cube with no previous configuration by a security administrator • The evaluator verifies the connection attempt with packet capture evidence. • The evaluator verifies the connection attempt with log evidence.
Pass/Fail with Explanation	The TOE does not allow another TOE component to be successfully connected since it was not configured.

6.1.2.2 FCO_CPC_EXT.1 TEST #1B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	(Note: Section 3.5.1 lists questions for which the evaluator shall determine and report answers through the combination of the TSS, Guidance Documentation, and Tests Evaluation Activities.) The evaluator shall perform the following tests: Test 1b: the evaluator shall confirm that after enablement, an IT entity can communicate only with the components that it has been enabled for. This includes testing that the enabled communication is successful for the enabled component pair, and that

	communication remains unsuccessful with any other component for which communication has not been explicitly enabled. Some TOEs may set up the registration channel before the enablement step is carried out, but in such a case the channel must not allow communications until after the enablement step has been completed. The evaluator shall repeat Tests 1a and 1b for each different type of enablement process that can be used in the TOE.
Test Steps	• The evaluator starts the enablement with a valid IT entity. • The evaluator connects the devices. • The evaluator verifies the connection attempt with packet capture evidence. • The evaluator verifies the connection attempt with log evidence.
Pass/Fail with Explanation	Pass. The TOE can only communicate with configured components and the channel was protected with IPSec.

6.1.2.3 FCO_CPC_EXT.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	(Note: Section 3.5.1 lists questions for which the evaluator shall determine and report answers through the combination of the TSS, Guidance Documentation, and Tests Evaluation Activities.) The evaluator shall perform the following tests: Test 2: The evaluator shall separately disable each TOE component in turn and ensure that the other TOE components cannot then communicate with the disabled component, whether by attempting to initiate communications with the disabled component or by responding to communication attempts from the disabled component.
Test Steps	• The evaluator disabled the TOE component. • The evaluator attempts a connection with the disabled TOE component and verify it fails. • The evaluator verifies with packet capture evidence that the devices did not connect due to enablement being disabled by the security administrator. • The evaluator verifies with log evidence the devices did not connect.
Pass/Fail with Explanation	The TOE components do not allow traffic to pass because one component of the TOE is disabled.

6.1.2.4 FCO_CPC_EXT.1 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	(Note: Section 3.5.1 lists questions for which the evaluator shall determine and report answers through the combination of the TSS, Guidance Documentation, and Tests Evaluation Activities.) The evaluator shall perform the following tests: Test 3: The evaluator shall perform the following tests according to those that apply to the values of the main (outer) selection made in the ST for FCO_CPC_EXT.1.2. i. If the ST uses the first type of communication channel in the selection in FCO_CPC_EXT.1.2 then the evaluator shall test the channel via the Evaluation Activities for FTP_ITC.1 or FPT_ITT.1 according to the second selection – the evaluator shall ensure that the test coverage for these SFRs includes their use in the registration process. ii. If the ST uses the second type of communication channel in the selection in FCO_CPC_EXT.1.2 then the evaluator shall test the channel via the Evaluation Activities for FTP_TRP.1/Join. iii. If the ST uses the 'no channel' selection, then no test is required.
Pass/Fail with Explanation	Pass. This test has been exercised in FPT_ITT.1 Test 1.

6.1.2.5 FCO_CPC_EXT.1 TEST #4 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	(Note: Section 3.5.1 lists questions for which the evaluator shall determine and report answers through the combination of the TSS, Guidance Documentation, and Tests Evaluation Activities.) The evaluator shall perform the following tests: Test 4: The evaluator shall perform one of the following tests, according to the TOE characteristics identified in its TSS and operational guidance: i. If the registration channel is not subsequently used for inter-component communication, and in all cases where the second selection in FCO_CPC_EXT.1.2 is made (i.e. using FTP_TRP.1/Join) then the evaluator shall confirm that the registration channel can no longer be used after the registration process has completed, by attempting to use the channel to communicate with each of the endpoints after registration has completed. ii. If the registration channel is subsequently used for intercomponent communication then the evaluator shall confirm that any aspects identified in the operational guidance as necessary to meet the requirements for a steady-state intercomponent channel (as in FTP_ITC.1 or FPT_ITT.1) can

	indeed be carried out (e.g. there might be a requirement to replace the default key pair and/or public key certificate).
Pass/Fail with Explanation	Pass. This test has been exercised in FPT_ITT.1 Test 1.

6.1.2.6 FCO_CPC_EXT.1 TEST #5 (CPP_ND_V3.0E)

Item	**Data**
Test Assurance Activity	(Note: Section 3.5.1 lists questions for which the evaluator shall determine and report answers through the combination of the TSS, Guidance Documentation, and Tests Evaluation Activities.) The evaluator shall perform the following tests: Test 5: For each aspect of the security of the registration channel that operational guidance states can be modified by the operational environment in order to improve the channel security (see AGD_PRE.1 refinement item 2 in (see the requirements on Preparative Procedures in 3.5.1.2), the evaluator shall confirm, by following the procedure described in the operational guidance, that this modification can be successfully carried out.
Pass/Fail with Explanation	Pass. The evaluator followed the guidance to perform all previous tests.

6.1.3 CRYPTOGRAPHIC SUPPORT (FCS)

6.1.3.1 FCS_CKM.1 RSA (CPP_ND_V3.0E) **[TD0921]**

Item	**Data**
Test Assurance Activity	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 or FIPS PUB 186-5 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. This test must be repeated for each supported RSA modulo and generation method. Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include: a) Random Provable Primes (p and q shall be provable primes) b) Random Probable primes (p and q shall be probable primes) c) Provable Primes with Conditions (p1, p2, q1, q2, p and q shall all be provable primes) d) Provable/probable primes with Conditions (p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes) e) Probable primes with Conditions (p1, p2, q1, q2, p and q shall all be probable primes) The Random Provable primes, and all the Primes with Conditions can be tested in the same manner because each of these begin with a starting random number and calculate the p and q values from this value. The test instructs the TSF to generate intermediate values and the p, q, n, and d values. The evaluator then validates the correctness of the values generated by the TSF. To test the key generation method for the Random Provable primes method or 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. If the TSF provides the input to the key generation function, such input must be recorded and verified. For each RSA key length (modulo) claimed, the evaluator shall generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing Key Pair values generated by the TSF with those generated using the same set of input values using a known good implementation. The Random Probable primes must be tested in a different way because this test generates different random numbers, not related to each other, until the number satisfies the "probably prime" requirements. This validation method requires two tests for Random Probable primes. These include the Known Answer Test and the Miller-Rabin probabilistic primality test. To test the key generation method for the Random Probable primes, the known answer test must be used. The evaluator shall compare the results of a known good implementation with the TSF results for a set (of a corresponding size depending on modulus) of supplied values containing private prime factor p, private prime factor

	q and confirm all public keys, e, match. Then the evaluator shall use the TSF to generate prime p, q pairs for each modulus size and perform the Miller-Rabin tests. **TD0921 has been applied.**
Pass/Fail with Explanation	RSA key generation is not claimed in the ST; hence, this activity is not applicable.

6.1.3.2 FCS_CKM.1 ECC (CPP_ND_V3.0E) **[TD0921]**

Item	Data
Test Assurance Activity	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 Elliptic Curve Cryptography (ECC)** *Key Generation for ECDSA Schemes* Key pairs for the ECDSA consist of pairs (d, Q), where the private key, d, is an integer, and the public key, Q, is an elliptic curve point. The evaluator shall verify the implementation of ECC Key Generation by the TOE using the following components tests: • ECC Key Pair Generation • ECC Public Key Validation (PKV) *ECC Key Pair Generation Test* There are four methods by which these pairs may be generated: • FIPS 186-4 Section B.4.1 Key Pair Generation Using Extra Random Bits Using this method, 64 more bits are requested from the RBG than are needed for d so that bias produced by the mod function is negligible. • FIPS 186-4 Section B.4.2 Key Pair Generation by Testing Candidates Using this method, a random number is obtained and tested to determine that it will produce a value of d in the correct range. If d is out-of-range, another random number is obtained (i.e., the process is iterated until an acceptable value of d is obtained. • FIPS 186-5 Section A.2.1 ECDSA Key Pair Generation using Extra Random Bits

	Using this method, more bits are requested from the DRBG than are needed for d so that the bias produced by the mod function is negligible. • FIPS 186-5 Section A2.2 ECDSA Key Pair Generation by Rejection Sampling Using this method, a random number is obtained and tested to determine that it will produce a value of d in the correct range. If d is out of range, an ERROR is returned. The evaluator shall test the ECC Key Pair Generation by having the TSF produce 10 key pairs (d, Q) for each implemented key generation method using each supported curve (i.e., P-256, P-384, and P-521). The steps are the same for each key generation method and curve. The private key, d, shall be generated using the output of an approved DRBG converted to an integer via modular reduction or the discard method. The known private key is then used by the TSF to compute the public key, Q'. To evaluator then validates the correctness by comparing the value Q' computed by the TSF to the public key, Q generated by a known good implementation. *ECC Public Key Verification (PKV) Test* The evaluator shall generate 12 key pairs (d, Q) for each selected curve, with 6 valid public keys, Q, using a known good implementation and 6 modified, Q', and determine whether the TSF can accurately detect these modifications. Q' should be otherwise valid but include at least one of the following errors: a) point X or Y not on the curve, b) point X or Y is too large for the field for the given curve. The evaluator encouraged to make sure that the modification does not inadvertently result in another valid public key (e.g., modifying Y and accidentally hitting a different point on the curve). **TD0921 has been applied.**
Pass/Fail with Explanation	Algorithm: ECDSA KeyGen (FIPS186-5) and ECDSA KeyVer (FIPS186-5) Curves: P-384, P-521 CAVP #: A7437 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.3 FCS_CKM.1 FFC - FIPS PUB 186-4 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 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.
Pass/Fail with Explanation	The DSA KeyGen is not claimed in the ST; hence, this activity is not applicable.

6.1.3.4 FCS_CKM.1 FFC - "SAFE-PRIME" GROUPS (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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). **FFC Schemes using "safe-prime" groups** Testing for FFC Schemes using safe-prime groups is done as part of testing in CKM.2.1.
Pass/Fail with Explanation	Safe-prime groups are not claimed in the ST; hence, this activity is not applicable.

6.1.3.5 FCS_CKM.1/IKE TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	**For FFC Schemes using "safe-prime" groups:** Testing for FFC Schemes using safe-prime groups is done as part of testing in FCS_CKM.2. **For all other selections:** The evaluator shall perform the corresponding tests for FCS_CKM.1 specified in the NDcPP SD, based on the selections chosen for this SFR. If IKE key generation is implemented by a different algorithm than the NDcPP key generation function, the evaluator shall ensure this testing is performed using the correct implementation.
Pass/Fail with Explanation	Pass. This test assurance is completed in conjunction with the FCS_CKM.1 RSA, FCS_CKM.1 ECC, FCS_CKM.1 FFC - "SAFE-PRIME" GROUPS in the NDcPPv3.0e TR.

6.1.3.6 FCS_CKM.2 RSA (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	**RSA-based key establishment** The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses RSAES-PKCS1-v1_5.

Pass/Fail with Explanation	RSA-based key establishment is not claimed in the ST; hence, this activity is not applicable.

6.1.3.7 FCS_CKM.2 SP800-56A - ECC (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	**Key Establishment Schemes** The evaluator shall verify the implementation of the key establishment schemes of the supported by the TOE using the applicable tests below. **ECC and FIPS 186-type FFC 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 for ECC and FIPS186- type. 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 shall 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.
Pass/Fail with Explanation	Algorithm: KAS_ECC_CDH-Component SP800-56Ar3 Curves: P-384, P-521 CAVP #: A7437 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.8 FCS_CKM.2 SP800-56A - FFC (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	**Key Establishment Schemes** The evaluator shall verify the implementation of the key establishment schemes of the supported by the TOE using the applicable tests below. **ECC and FIPS 186-type FFC 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 for ECC and FIPS186- type. 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 shall 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.
Pass/Fail with Explanation	FIPS 186-type FFC SP800-56A Key Establishment Schemes are not claimed in the ST; hence, this activity is not applicable.

6.1.3.9 FCS_CKM.2 FFC SAFE-PRIME (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	**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.
Pass/Fail with Explanation	FFC Schemes using “safe-prime” groups are not claimed in the ST; hence, this activity is not applicable.

6.1.3.10 FCS_CKM.4 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	There are no test assurance activities.
Pass/Fail with Explanation	N/A. There are no test assurance activities for this SFR. Based on these findings, this assurance activity is considered satisfied.

6.1.3.11 FCS_COP.1/DATAENCRYPTION TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	There is no change to the Evaluation Activities specified for this SFR in the NDcPP Supporting Document. The PP-Module modifies this SFR to require the ST author to make certain selections, but these selections are all part of the original definition of the SFR so no new behavior is defined by the PP-Module.
Pass/Fail with Explanation	Pass. This test is covered by the Crypto module in the NDcPPv3.0e TR.

6.1.3.12 FCS_COP.1/DATAENCRYPTION AES-CBC (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	**AES-CBC Known Answer Tests** There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation. **KAT-1.** To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros.

	Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all-zeros key. To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption. **KAT-2.** To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys. To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using an all-zero ciphertext value as input and AES-CBC decryption. **KAT-3**. To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. To test the decrypt functionality of AES-CBC, the evaluator shall supply the two sets of keys and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. The ciphertext value in each pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key. **KAT-4.** To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256- bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128].

	To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption.
	
	AES-CBC Multi-Block Message Test
	The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation.
	
	The evaluator shall also test the decrypt functionality for each mode by decrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.
	
	AES-CBC Monte Carlo Tests
	The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:
	
	# Input: PT, IV, Key
	for i = 1 to 1000:
	  if i == 1:
	  CT[1] = AES-CBC-Encrypt(Key, IV, PT)
	  PT = IV
	  else:
	  CT[i] = AES-CBC-Encrypt(Key, PT)
	  PT = CT[i-1]
	
	The ciphertext computed in the 1000th iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.
	
	The evaluator shall test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AESCBC-Decrypt.

Pass/Fail with Explanation	The AES-CBC algorithm is not claimed in the ST; hence, this activity is not applicable.

6.1.3.13 FCS_COP.1/DATAENCRYPTION AES-GCM (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	**AES-GCM Test** The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths: **128 bit and 256 bit keys** a) **Two plaintext lengths.** One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported. a) **Three AAD lengths.** One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported. b) **Two IV lengths.** If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested. The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known. The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.
Pass/Fail with Explanation	Algorithm: AES GCM Key size: 256 CAVP #: A7437 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.14 FCS_COP.1/DATAENCRYPTION AES-CTR (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	**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 if the TSF is validated against the requirements of the Functional Package for Secure Shell referenced in Section 2.2 of the cPP. If CBC and/or GCM are selected in FCS_COP.1/DataEncryption, the test activities for those modes sufficiently demonstrate the correctness of the AES algorithm. If CTR is the only selection in FCS_COP.1/DataEncryption, the AES-CBC Known Answer Test, AES-GCM Known Answer Test, or the following test shall be performed (all of these tests demonstrate the correctness of the AES algorithm): There are four Known Answer Tests (KATs) described below to test a basic AES encryption operation (AES-ECB mode). For all KATs, the plaintext, ~~IV~~, and ciphertext values shall be 128-bit blocks. The results from each test may either be obtained by the validator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation. **KAT-1** To test the encrypt functionality, the evaluator shall supply a set of 5 plaintext values for each selected keysize and obtain the ciphertext value that results from encryption of the given plaintext using a key value of all zeros. **KAT-2** To test the encrypt functionality, the evaluator shall supply a set of 5 key values for each selected keysize and obtain the ciphertext value that results from encryption of an all zeros plaintext using the given key value. **KAT-3** To test the encrypt functionality, the evaluator shall supply a set of key values for each selected keysize as described below and obtain the ciphertext values that result from AES encryption of an all zeros plaintext using the given key values. A set of 128 128-bit keys, a set of 192 192-bit keys, and/or a set of 256 256-bit keys. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1, N].

	KAT-4 To test the encrypt functionality, the evaluator shall supply the set of 128 plaintext values described below and obtain the ciphertext values that result from encryption of the given plaintext using each selected keysize with a key value of all zeros (e.g. 256 ciphertext values will be generated if 128 bits and 256 bits are selected and 384 ciphertext values will be generated if all keysizes are selected). Plaintext value i in each set shall have the leftmost bits be ones and the rightmost 128-i bits be zeros, for i in [1, 128]. **AES-CTR Multi-Block Message Test** The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10 (test shall be performed using AES-ECB mode). For each i the evaluator shall choose a key and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key using a known good implementation. The evaluator shall perform this test using each selected keysize. **AES-CTR Monte-Carlo Test** The evaluator shall test the encrypt functionality using 100 plaintext/key pairs. The plaintext values shall be 128-bit blocks. For each pair, 1000 iterations shall be run as follows: # Input: PT, Key for i = 1 to 1000: CT[i] = AES-ECB-Encrypt(Key, PT) PT = CT[i] The ciphertext computed in the 1000th iteration is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation. The evaluator shall perform this test using each selected keysize. There is no need to test the decryption engine.
Pass/Fail with Explanation	The AES CTR algorithm is not claimed in the ST; hence, this activity is not applicable.

6.1.3.15 FCS_COP.1/SIGGEN ECDSA (CPP_ND_V3.0E) **[TD0921]**

Item	**Data**
Test Assurance Activity	**ECDSA Signature Algorithm Tests**

	The evaluator shall verify the implementation of ECDSA Signature generation by the TOE using the Signature Generation Test. This test validates the TSF generation of the (r, s) pair that represent the digital signature. The digital signature shall be verified using the same domain parameters and hash function that were used during signature generation. An approved hash function or an XOF shall be used for this purpose. **Signature Generation Test** The purpose of this test is to verify the ability of the TSF to produce correct signatures. To test signature generation, the evaluator supplies 10 pseudorandom messages to the TSF and a key pair, (d, Q), generated by a known good implementation. Exercising each applicable curve (i.e., P-256, P 384, or P-521) and hash algorithm or extendable-output function combination, the TSF generates signatures for each supplied message and returns the corresponding signatures. Using a known-good implementation, the evaluator validates the signatures by using the associated public key, Q, to verify the signature. **Signature Verification Test** The purpose of this test is to verify the ability of the TSF to accept valid signatures and reject invalid signatures. For each curve/hash algorithm or extendable-output function combination supported by the TSF, the evaluator shall use a known good implementation to generate a key pair, (d, Q), and use known good implementation with the private key, d, to sign 15 pseudorandom messages of 1024 bits. The evaluator shall alter some of the messages or signatures so that signature verification should fail. To test signature verification, the evaluator supplies the public key, Q, and 15 pseudorandom messages, including altered messaged, to the TSF for verification of signatures. The evaluator shall then verify that the TSF validates correct signatures on the original messages and flags or rejects the altered messages. **TD0921 has been applied.**
Pass/Fail with Explanation	Algorithm: ECDSA SigGen, ECDSA SigVer Curves: P-384, P-521 CAVP #: A7437 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.16 FCS_COP.1/SIGGEN RSA (CPP_ND_V3.0E) **[TD0921]**

Item	Data
Test Assurance Activity	**RSA Signature Algorithm Tests** **Signature Generation Test** The evaluator shall verify the implementation of RSA Signature generation by the TOE using the Signature Generation Test. This test verifies the ability of the TSF to produce correct signatures. There are 2 different RSA Signature algorithms that can be implemented. These include: a. RSASSA-PKCS1-v1.5 b. RSASSA-PSS To test signature generation, the evaluator generates or obtains 10 messages for each modulus size/hash or extendable-output function combination supported by the TOE. Using a key generated by a known good implementation, the TSF generates and returns the corresponding signatures to a known good implementation that validates the signatures by using the associated public key to verify the signature. 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** The evaluator shall verify the implementation of RSA Signature verification by the TOE using the Signature Verification Test. This test verifies the ability of the TSF to recognize valid and invalid signatures. There are 2 different RSA Signature algorithms that can be implemented. These include: a. RSASSA-PKCS1-v1.5 b. RSASSA-PSS For each modulus size/hash or extendable-output function combination supported by the TOE, the evaluator shall use a known good implementation to generate a modulus and three associated key pairs, (d, e). Each private key d is used to sign six pseudorandom messages each of 1024 bits. Some of the public keys, e, messages, IR format, or signatures must be altered so that signature verification should fail. The modifications must cover distinct "key modified", "message modified", "signature modified", "IR moved", and "trailer moved" tests. The modulus, hash or

	extendable-output algorithm, public key e values, messages, and signatures are forwarded to the TSF. The TSF then attempts to verify the signatures and returns the results. The evaluator then compares the received results with the stored results from a known good implementation. The evaluator verifies that the TSF validates correct signatures on the original messages and flags or rejects the altered messages. **TD0921 has been applied.**
Pass/Fail with Explanation	RSA Signature Algorithm is not claimed in the ST; hence, this activity is not applicable.

6.1.3.17 FCS_COP.1/HASH (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The TSF hashing functions can be implemented in one of two modes. The first mode is the byteoriented mode. In this mode the TSF 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 bitoriented 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 bitoriented vs. the byteoriented 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 evaluator shall 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 evaluator shall 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 evaluator shall 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 evaluator shall 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 byteoriented implementations only. The evaluator shall 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 evaluator shall then ensure that the correct result is produced when the messages are provided to the TSF.
Pass/Fail with Explanation	Algorithm: SHA-256, SHA-384, SHA-512 CAVP #: A7437 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.18 FCS_COP.1/KEYEDHASH (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and message data using a known good implementation.
Pass/Fail with Explanation	Algorithm: HMAC-SHA2-384, HMAC-SHA2-512 CAVP #: A7437 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.19 FCS_HTTPS_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	This test is now performed as part of FIA_X509_EXT.1/Rev testing. Tests are performed in conjunction with the TLS evaluation activities. If the TOE is an HTTPS client or an HTTPS server utilizing X.509 client authentication, then the certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.
Pass/Fail with Explanation	Pass. This test has been exercised with FIA_X509_EXT.1/Rev testing.

6.1.3.20 FCS_IPSEC_EXT.1 TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	There are no additional testing activities.
Pass/Fail with Explanation	NA. There are no additional testing activities.

6.1.3.21 FCS_IPSEC_EXT.1.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall use the guidance documentation to configure the TOE to perform the following tests: 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 shall perform 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 shall observe 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	Protect (Positive Case): • Configure an access rule to protect the ICMP traffic. • Start the IPsec connection and generate traffic. • Verify that the ICMP packets are allowed. • Verify via packet capture that the packets are sent encrypted.

	Note- Negative Case: This is covered by the "deny" sub-test below. Deny (Positive Case): • Configure an access list to deny the ICMP traffic. • Start the IPsec connection and generate traffic. • Verify that the ICMP packets are denied. • Verify via packet capture that the ICMP packets are dropped. Note- Negative Case: Covered by the "protect" (above) and "bypass" sub-test (below). Bypass (Positive Case): • Configure an access list to bypass the traffic. • Generate traffic to match the configured rule. • Verify that the ICMP packets are bypassed. • Verify via packet capture that the traffic is bypassed.
Pass/Fail with Explanation	Pass. The TOE dropped packets when configured, encrypted packets when configured, and sent packets in plaintext when configured. This meets the testing requirements.

6.1.3.22 FCS_IPSEC_EXT.1.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall use the guidance documentation to configure the TOE to perform the following tests: Test 2: The evaluator shall devise several tests that cover a variety of scenarios for packet processing. As with Test 1, the evaluator shall ensure both positive and negative test cases are constructed. These scenarios must exercise the range of possibilities for SPD entries and processing modes as outlined in the TSS and guidance documentation. Potential areas to cover include rules with overlapping ranges and conflicting entries, inbound and outbound packets, and packets that establish SAs as well as packets that belong to established SAs. The evaluator shall verify, via the audit trail and packet captures, for each scenario that the expected behavior is exhibited, and is consistent with both the TSS and the guidance documentation.
Test Steps	Add conflicting rules: Covered in test case FFW_RUL_EXT.1.8 TEST #1. Protect a large set and deny a small subset:

	• Configure the device to discard a small subset of traffic (ICMP) and protect a large set of traffic. • Ping from the PEER and verify the ICMP request fails, and all other traffic is allowed. • Verify that only the ICMP has been dropped. • Verify that all other traffic is allowed. • Change the priorities for the created rules. • Generate traffic to match the configured rules. Verify that ICMP packets are allowed. • Verify via packet capture that the ICMP packets and all other traffic are allowed.
Pass/Fail with Explanation	Pass. The TOE was able to successfully process the traffic according to the configured rules, even with overlapping ranges and conflicting entries. This meets the testing requirements.

6.1.3.23 FCS_IPSEC_EXT.1.2 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The assurance activity for this element is performed in conjunction with the activities for FCS_IPSEC_EXT.1.1. The evaluator shall use the guidance documentation to configure the TOE to perform the following tests: 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 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 shall 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 shall send the packet and observe that the packet was dropped.
Test Steps	• Configure the SPD such that there is a rule for dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext • Construct a network packet that matches the rule to allow the packet to flow in plaintext and send that packet • Observe that the network packet is passed to the proper destination interface with no modification

	• Modify a field in the packet header; such that it no longer matches the evaluator-created entries • Send the packet and observe that the packet was dropped
Pass/Fail with Explanation	Pass. The TOE allows packets that meet the SPD ruleset for BYPASS and does not allow packets that do not meet the SPD ruleset.

6.1.3.24 FCS_IPSEC_EXT.1.3 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following test(s) based on the selections chosen: Test 1: If **tunnel mode** is selected, the evaluator shall use 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 shall configure 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 shall observe (for example, in the audit trail and the captured packets) that a successful connection was established using the tunnel mode.
Test Steps	• Configure the TOE and peer to establish an IPsec connection. • Initiate traffic through the IPSec tunnel between the TOE and the peer. • Verify that the traffic is encrypted with IPsec via PCAP. • Verify successful IPsec tunnel establishment via logs and that tunnel mode is used.
Pass/Fail with Explanation	Pass. The TOE can be configured to operate in tunnel mode and successfully establish a connection with a peer using this mode. This meets the testing requirements.

6.1.3.25 FCS_IPSEC_EXT.1.3 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following test(s) based on the selections chosen: Test 2: If **transport mode** is selected, the evaluator shall use 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 shall configure 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 shall then initiate a connection from the TOE to connect to the VPN peer. The evaluator shall observe (for example, in the audit trail and the captured packets) that a successful connection was established using the transport mode.
Pass/Fail with Explanation	NA. This test is not applicable because transport mode selection is not included in the ST.

6.1.3.26 FCS_IPSEC_EXT.1.4 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure the TOE as indicated in the guidance documentation configuring the TOE to use each of the supported algorithms, attempt to establish a connection using ESP, and verify that the attempt succeeds.
Test Steps	Note: MAC is implicit for GCM encryption and need not be configured. • Configure the TOE for AES-GCM-256 in ESP. • Configure the PEER for AES-GCM-256 in ESP. • Initiate traffic through the IPSec tunnel between the TOE and the peer. • Verify the successful connection using packet capture. • Verify using TOE logs that the connection was established using AES-GCM-256.
Pass/Fail with Explanation	Pass. The TOE can be configured with each supported algorithm and successfully negotiates and establishes a secure connection with the peer using these ESP algorithms. This meets the testing requirements.

6.1.3.27 FCS_IPSEC_EXT.1.5 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Tests are performed in conjunction with the other IPsec evaluation activities. 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 shall then show that main mode exchanges are supported.
Pass/Fail with Explanation	N/A. This test is not applicable since 'IKEv1' selection is not included in the ST.

6.1.3.28 FCS_IPSEC_EXT.1.5 TEST #2 (CPP_ND_V3.0E)

Item	Data

Test Assurance Activity	Tests are performed in conjunction with the other IPsec evaluation activities. 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 7296, Section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.
Test Steps	• The Evaluator attempts an IPSec connection between the GoSilent Server and GoSilent Cube. • The evaluator verifies the successful connection using a packet capture. • Verify the successful connection using TOE logs.
Pass/Fail with Explanation	Pass. The TOE successfully performs NAT traversal processing. The TOE logs and packet capture shows that TOE negotiated the VPN tunnel with NAT traversal.

6.1.3.29 FCS_IPSEC_EXT.1.6 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 shall confirm the algorithm was that used in the negotiation.
Test Steps	• Configure the TOE with an IKEv2 policy using AES-GCM-256. • Configure the Peer with an IKEv2 policy using AES- GCM-256. • Initiate traffic through the IPSec tunnel between the TOE and the peer. • Verify the successful connection using packet capture. • Verify the successful connection using TOE logs.
Pass/Fail with Explanation	Pass. The TOE establishes a connection with the peer device, and the connection is maintained without errors or fallback to a different ciphersuite. This indicates that the TOE is correctly using the specified ciphersuite for encrypting the IKEv2 payload. This meets the testing requirements.

6.1.3.30 FCS_IPSEC_EXT.1.7 TEST #1 **[TD0868] (CPP_ND_V3.0E)**

Item	Data
Test Assurance Activity	When testing this functionality, the evaluator shall ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is

	responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered." Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection: Test 1: If '**number of bytes'** is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime in terms of the number of bytes allowed following the guidance documentation. The evaluator shall configure a test peer with a byte lifetime that exceeds the lifetime of the TOE. The evaluator shall establish a SA between the TOE and the test peer, and determine that once the allowed number of bytes through this SA is exceeded, a new SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 1 negotiation. **TD0868 has been applied.**
Pass/Fail with Explanation	NA. **Number of bytes** is not selected as the SA lifetime measure.

6.1.3.31 FCS_IPSEC_EXT.1.7 TEST #2 [TD0868] (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	When testing this functionality, the evaluator shall ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered." Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection: Test 2: If '**length of time'** is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime no later than 24 hours for the Phase 1 SA following the

	guidance documentation. The evaluator shall configure a test peer with a Phase 1 SA lifetime that exceeds the Phase 1 SA lifetime on the TOE. 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. **TD0868 has been applied.**
Test Steps	IKEv2 • Configure the TOE to support lifetime slightly lesser than 24 hours for IKEv2 Phase 1. • Configure the Peer to support 24 hours lifetime (86400 sec) for IKEv2 Phase 1. • Establish and maintain sessions for 24 hours. • Verify that the Rekey has happened before 24 hours via logs. • Verify that Phase 1 renegotiates before 24 hours via packet capture.
Pass/Fail with Explanation	Pass. The TOE correctly initiates a new Phase 1 SA negotiation and establishes it within the 24-hour timeframe. This meets the testing requirements.

6.1.3.32 FCS_IPSEC_EXT.1.8 TEST #1 **[TD0868] (CPP_ND_V3.0E)**

Item	Data
Test Assurance Activity	When testing this functionality, the evaluator shall ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered." Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection: Test 1: If '**number of bytes'** is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime in terms of the number of bytes allowed following the guidance documentation. The evaluator shall configure a test peer with a byte lifetime that exceeds the lifetime of the TOE. The evaluator shall establish a SA between the TOE and the test peer, and determine that once the allowed number of bytes through this SA is exceeded, a new SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 2 negotiation.

	TD0868 has been applied.
Pass/Fail with Explanation	NA. **Number of bytes** is not selected as the SA lifetime measure.

6.1.3.33 FCS_IPSEC_EXT.1.8 TEST #2 **[TD0868] (CPP_ND_V3.0E)**

Item	Data
Test Assurance Activity	When testing this functionality, the evaluator shall ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered." Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection: Test 2: If **'length of time'** is selected as the SA lifetime measure, the evaluator shall configure a maximum lifetime no later than 8 hours for the Phase 2 SA following the guidance documentation. The evaluator shall configure a test peer with a Phase 2 SA lifetime that exceeds the Phase 2 SA lifetime on the TOE. 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. **TD0868 has been applied.**
Test Steps	IKEv2: • Configure the IKEv2 Phase 2 SA Lifetime as 8 hours (28800 seconds) on the TOE. • Configure the IKEv2 Phase 2 SA lifetime for more than 8 hours (30000 seconds) on the peer. • Establish and maintain an IPsec connection between the TOE and peer for 8 hours. • Verify that a Phase 2 rekey was initiated before 8 hours via log review and packet capture.

Pass/Fail with Explanation	Pass. The TOE successfully initiates a Phase 2 negotiation before the 8-hour mark, confirming that it adheres to the configured maximum SA lifetime. This meets the testing requirements.

6.1.3.34 FCS_IPSEC_EXT.1.9 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	No test assurance activities have been identified.
Pass/Fail with Explanation	NA. No test assurance activities have been identified.

6.1.3.35 FCS_IPSEC_EXT.1.10 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection: Test 1: If 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.
Pass/Fail with Explanation	Pass. Covered by TSS Assurance Activities in the AAR.

6.1.3.36 FCS_IPSEC_EXT.1.10 TEST #2 (CPP_ND_V3.0E)

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

Pass/Fail with Explanation	Pass. Covered by TSS Assurance Activities in the AAR.

6.1.3.37 FCS_IPSEC_EXT.1.11 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	For each supported DH group, the evaluator shall test to ensure that all supported IKE protocols can be successfully completed using that particular DH group.
Test Steps	IKEv2: • Configure DH group 20 for IKEv2 on TOE. • Configure DH group 20 for IKEv2 on PEER. • Initiate traffic through the IPSec tunnel between the TOE and the peer. • Verify the successful connection using packet capture. • Verify the successful connection using TOE logs. • Configure DH group 21 for IKEv2 on TOE. • Configure DH group 21 for IKEv2 on PEER. • Initiate traffic through the IPSec tunnel between the TOE and the peer. • Verify the successful connection using packet capture. • Verify the successful connection using TOE logs.
Pass/Fail with Explanation	Pass. The successful completion of IKE protocols using each DH group confirms that the TOE correctly implements and supports all configured DH groups, ensuring secure key exchanges. This meets the testing requirements.

6.1.3.38 FCS_IPSEC_EXT.1.12 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall follow the guidance to configure the TOE to perform the following tests. 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.
Pass/Fail with Explanation	Pass. This test has been completed as part of testing covered in FCS_IPSEC_EXT.1.4 Test #1 and FCS_IPSEC_EXT.1.6 Test #1.

6.1.3.39 FCS_IPSEC_EXT.1.12 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall follow the guidance to configure the TOE to perform the following tests. 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.
Test Steps	IKEv2: • Configure the TOE with AES-GCM-128 in phase 1 and AES-GCM-256 in phase 2. • Configure peer to use AES-GCM-128 in phase 1 and AES-GCM-256 in phase 2. • Attempt to establish a connection. • Verify the connection is rejected using logs. • Verify the connection is rejected using Packet Capture.
Pass/Fail with Explanation	Pass. When attempting to connect to a peer with an IPsec SA strength larger than the IKE SA strength, the TOE rejects the connection for IKEv2. This meets the testing requirements.

6.1.3.40 FCS_IPSEC_EXT.1.12 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall follow the guidance to configure the TOE to perform the following tests. 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	IKEv2: • Configure the TOE to use AES-GCM-256. • Configure the Peer to use AES-GCM-128. • Attempt a secure IPsec connection. • Verify that the connection is rejected via packet capture. • Verify the logs reflected the failure on the TOE.
Pass/Fail with Explanation	Pass. The TOE supports and proposes the configured IKE SA algorithm. If the IPsec peer does not have matching algorithms the IPsec session is not established for IKEv2. This meets the testing requirements.

6.1.3.41 FCS_IPSEC_EXT.1.12 TEST #4 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall follow the guidance to configure the TOE to perform the following tests. 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	IKEv2:
	• Configure the TOE to support the following algorithms:
	  ○ IKEv2 SA (Phase 1): AES-GCM-256.
	  ○ IPsec SA (Phase 2): AES-GCM-256.
	• Configure a peer to support the following algorithms:
	  ○ IKEv2 SA (Phase 1): AES-GCM-256.
	  ○ IPsec SA (Phase 2): AES-GCM-128.
	• Attempt a secure IPsec connection.
	• Verify that the connection is rejected via packet capture.
	• Verify the logs reflected the failure on the TOE.

Pass/Fail with Explanation	Pass. Since the IPsec SA parameters of the TOE did not match that of the peer, an IPsec connection could not be established for IKEv2. An IKE SA, however, could be established because the peer parameters matched. This meets the testing requirements.

6.1.3.42 FCS_IPSEC_EXT.1.13 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. 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.

6.1.3.43 FCS_IPSEC_EXT.1.14 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	NA. No CN identifier type has been selected.

6.1.3.44 FCS_IPSEC_EXT.1.14 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 2 [conditional]: For each SAN/identifier type combination selected, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the field in the peer's presented certificate and shall verify that the IKE authentication succeeds. If the TOE prioritizes SAN checking over CN (through explicit specification of the field when specifying the reference identifier or prioritization rules), the evaluator shall also configure the CN so it contains an incorrect identifier formatted to be the same type (e.g. the reference identifier on the TOE is DNS-ID; identify certificate has an identifier in SAN with correct DNS-ID, CN with incorrect DNS-ID (and not a different type of identifier)) and verify that IKE authentication succeeds.
Pass/Fail with Explanation	NA. No SAN identifier type has been selected.

6.1.3.45 FCS_IPSEC_EXT.1.14 TEST #3 (CPP_ND_V3.0E)

Item	Data

Test Assurance Activity	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 3 [conditional]: For each CN/identifier type combination selected, the evaluator shall: i. 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. ii. 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.
Pass/Fail with Explanation	NA. No CN identifier type has been selected.

6.1.3.46 FCS_IPSEC_EXT.1.14 TEST #4 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 4 [conditional]: For each SAN/identifier type combination selected, the evaluator shall: i. 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. ii. 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.

Pass/Fail with Explanation	NA. No SAN identifier type has been selected.

6.1.3.47 FCS_IPSEC_EXT.1.14 TEST #5 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 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	• Configure the TOE's peer reference identifier to match the DN identifier fields presented in the peer certificate. • Establish a connection via the IPSec tunnel and observe that it is successful. • Verify that the connection is successful via TOE logs. • Verify that the connection is successful via packet capture.
Pass/Fail with Explanation	Pass. The TOE successfully establishes IKE authentication when the subject DN in the peer's certificate matches the configured reference identifier on the TOE, demonstrating that the TOE correctly handles DN identifier types during authentication.

6.1.3.48 FCS_IPSEC_EXT.1.14 TEST #6A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 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: i. Duplicate the CN field, so the otherwise authorized DN contains two identical CNs.
Test Steps	• Configure the peer reference identifier on the TOE to be the correct DN with a single CN. • Create a peer certificate with a single CN field. • Use the Acumen x509-mod tool to duplicate CN on the DN of the peer certificate. • Attempt a connection via the IPSec tunnel and observe that it fails. • Verify that the connection is not established via the logs. • Verify that IKE authentication fails via packet capture.
Pass/Fail with Explanation	Pass. The TOE correctly rejects the IKE authentication attempt when the DN contains duplicate CN fields, demonstrating that the TOE performs a strict bit-wise comparison of the DN and does not accept certificates with duplicated or non-compliant DN structures. This meets the testing requirements.

6.1.3.49 FCS_IPSEC_EXT.1.14 TEST #6B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 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: ii. Append '\0' to a non-CN field of an otherwise authorized DN.
Test Steps	• Configure the peer reference identifier on the TOE to be the correct DN. • Create a normal peer certificate. • Use the x509-mod tool to append '\0' to a non-CN field. • Configure Peer with the modified certificate. • Initiate an IPsec connection and verify that the IKE authentication fails. • Verify that the connection is not established via the logs. • Verify that IKE authentication fails via packet capture.

Pass/Fail with Explanation	Pass. The TOE correctly rejects the IKE authentication attempt when the DN includes a null character appended to a non-CN field. This demonstrates that the TOE performs a strict bit-wise comparison of the DN and does not accept certificates with non-standard or improperly formatted DN fields. This meets the testing requirements.

6.1.3.50 FCS_RBG_EXT.1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 shall verify 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 shall verify 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.
Pass/Fail with Explanation	Algorithm: CTR_DRBG Mode: AES-256 CAVP #: A7437 Pass. Based on these findings, this assurance activity is considered satisfied.

6.1.3.51 FCS_TLSC_EXT.1.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Steps	TLSv1.2: • Establish a connection with the TOE over TLS using the ciphersuite TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384. • Verify through packet capture that the required cipher is used in a successful TLS connection. TLSv1.3: • Establish a connection with the TOE over TLS using the ciphersuite TLS_AES_256_GCM_SHA384. • Verify through packet capture that the required cipher is used in a successful TLS connection.
Pass/Fail with Explanation	Pass. TOE successfully negotiates each of the claimed ciphersuites. This meets the test requirements.

6.1.3.52 FCS_TLSC_EXT.1.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: The goal of the following test is to verify that the TOE accepts only certificates with appropriate values in the extendedKeyUsage extension, and implicitly that the TOE correctly parses the extendedKeyUsage extension as part of X.509v3 server certificate validation. Test 2: The evaluator shall establish the connection with a server presenting a certificate that contains the serverAuth (OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage extension and verify that the connection successfully negotiated. The evaluator shall then verify that when the same server presents an otherwise valid server certificate that contains the extendedKeyUsage extension without serverAuth the client rejects the connection. Ideally, the two certificates should be identical except for the OID values.
Test Steps	Certificate with server EKU present: • Create the server certificate containing the Server Authentication purpose. • Try to establish a connection with the TOE over TLS using the certificate created and verify that it is successful. • Verify the successful connection with packet capture. Certificate with server EKU absent: • Create the server certificate lacking the Server Authentication purpose. • Try to establish a connection with the TOE over TLS using the certificate created and verify that it is unsuccessful. • Verify the error logs on the device showing the connection is rejected due to an invalid extension in the certificate. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. TOE successfully established the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and TOE rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field. This meets the test requirements.

6.1.3.53 FCS_TLSC_EXT.1.1 TEST #3 (CPP_ND_V3.0E)

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

	Test 3 [conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall send a server certificate in the TLS connection that does not match the server-selected ciphersuite (for example, send an ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite). The evaluator shall verify that the TOE disconnects after receiving the server's Certificate handshake message.
Test Steps	• Start the server using the 'acumen-tls' tool with a certificate that does not match the server-selected ciphersuite and verify that the connection attempt fails. • Verify the device logs for errors indicating a cipher mismatch. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. The TOE denied a connection to a server using a certificate that doesn't match the ciphersuite. This meets the test requirements.

6.1.3.54 FCS_TLSC_EXT.1.1 TEST #4A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 4: The evaluator shall perform the following 'negative tests': i. [conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the TOE TLS client denies the connection.
Test Steps	• Start the server using the 'acumen-tls' and send a server hello selecting TLS_NULL_WITH_NULL NULL cipher suite. Verify that the connection attempt fails. • Verify that the error logs on the device are showing failure due to an unknown cipher. • Verify the handshake failure using packet capture.
Pass/Fail with Explanation	Pass. The TOE denies the session because TLS_NULL_WITH_NULL_NULL is presented. This meets the test requirements.

6.1.3.55 FCS_TLSC_EXT.1.1 TEST #4B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 4: The evaluator shall perform the following 'negative tests': ii. Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite (compatible with the server-selected version

	of TLS) not presented in the Client Hello handshake message. The evaluator shall verify that the TOE TLS client rejects the connection after receiving the Server Hello.
Test Steps	• Start the server using the 'acumen-tls' tool and modify the Server Hello message to include a cipher suite that was not offered in the Client Hello message. Verify that the connection attempt is rejected. • Check the device logs for error messages indicating a connection failure due to an unsupported cipher suite. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects the connection after receiving a server hello that includes a cipher suite not offered in the client hello, by sending a Fatal Alert. This meets the test requirements.

6.1.3.56 FCS_TLSC_EXT.1.1 TEST #4C (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 4: The evaluator shall perform the following 'negative tests': iii. The evaluator shall attempt to establish a TLS connection using each valid TLS/SSL version (i.e. TLS 1.3, TLS 1.2, TLS 1.1, TLS 1.0, SSL 3.0, SSL 2.0). The evaluator shall verify that the version(s) specified in FCS_TLSC_EXT.1.1 are successfully established and all other versions are rejected by the TOE TLS client. If a supported_versions extension is not sent by the TOE in the ClientHello, then the evaluator must ensure the test server responds with a ServerHello that is valid for the TLS version being negotiated. If the TOE includes the Supported Versions extension in its ClientHello, the evaluator shall also ensure the version(s) specified in the extension match the version(s) in FCS_TLSC_EXT.1.1. NOTE: For TLS 1.3 aware test servers, it is appropriate for the test server to issue a TLS Alert. The TOE client must not attempt to continue the connection.
Test Steps	TLSv1.2 • Establish a connection with the TOE over TLS and verify that the connection with the TOE is successfully established using TLS v1.2. • Confirm that the connection was successful by analyzing the packet capture. TLSv1.3 • Establish a connection with the TOE over TLS and verify that the connection with the TOE is successfully established using TLS v1.3. • Confirm that the connection was successful by analyzing the packet capture.

	SSL v2.0 • Start the server using the 'acumen-tls' tool and verify that the connection with the TOE using SSL v2.0 fails. • Verify the connection failure using the device logs. • Verify the TLS handshake failure by analyzing the packet capture. SSL v3.0 • Start the server using the 'acumen-tls' tool and verify that the connection with the TOE using SSL v3.0 fails. • Verify the connection failure using the device logs. • Verify the TLS handshake failure by analyzing the packet capture. TLS v1.0 • Start the server using the 'acumen-tls' tool and verify that the connection with the TOE using TLS v1.0 fails. • Verify the connection failure using the device logs. • Verify the TLS handshake failure by analyzing the packet capture. TLS v1.1 • Start the server using the 'acumen-tls' tool and verify that the connection with the TOE using TLS v1.1 fails. • Verify the connection failure using the device logs. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects all SSLv2, SSLv3, TLS v1.0, and TLS v1.1 connection attempts and accepts the connection when TLSv1.2 and TLSv1.3 is used. This meets the testing requirement.

6.1.3.57 FCS_TLSC_EXT.1.1 TEST #5A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 5: The evaluator shall perform the following modifications to the traffic (i.e. Man-in-the-middle modifications that result in invalid signatures and MACs): i. [conditional]: Perform this test only if support of TLS 1.2 is claimed. If using DHE or ECDH ciphersuites, modify the signature block in the Server's Key Exchange handshake message, and verify that the client denies the connection and no application data flows. The handshake shall be valid (e.g. the Finished message is calculated using the modified

	signature), with the exception of the invalid signature. This test does not apply to ciphersuites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.
Test Steps	• Start the server using the 'acumen-tls' tool and verify that the TOE rejects the connection when a signature byte is modified in the Server's Key Exchange handshake message. • Verify the error logs on the device indicating connection failure due to a bad signature. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects the connection due to the modified signature block in the Server Key Exchange message. This meets the test requirement.

6.1.3.58 FCS_TLSC_EXT.1.1 TEST #5B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 5: The evaluator shall perform the following modifications to the traffic (i.e. Man-in-the-middle modifications that result in invalid signatures and MACs): ii. [conditional]: Perform this test only if support of TLS 1.3 is claimed. Modify the signature block in the Server's Certificate Verify handshake message, and verify that the client denies the connection and no application data flows. The handshake shall be valid (e.g. the Finished message is calculated using the modified signature), with the exception of the invalid signature.
Test Steps	• Start the server using the 'acumen-tls' tool and verify that the TOE rejects the connection when the signature block in the Server's Certificate Verify handshake message is modified. • Verify the error logs on the device showing connection failure due to a bad signature. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when the signature block in the Server's Certificate Verify handshake message is modified. This meets the test requirement.

6.1.3.59 FCS_TLSC_EXT.1.1 TEST #6A (CPP_ND_V3.0E)

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

	Test 6: The evaluator shall perform the following 'scrambled message tests': i. Modify a byte in the Server Finished handshake message and verify that the handshake does not finish successfully and no application data flows. (Note: This modification must be performed prior to the contents of the Finished message being encrypted.)
Test Steps	• Start the server using the 'acumen-tls' tool and verify that the TOE rejects the connection when a byte is modified in the server finished handshake. • Verify the error logs on the device indicating decryption failure. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. When a byte is modified in the Server Finished handshake message, the handshake does not complete successfully, and no application data is transmitted. This meets the test requirements.

6.1.3.60 FCS_TLSC_EXT.1.1 TEST #6B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 6: The evaluator shall perform the following 'scrambled message tests': ii. [conditional]: Perform this test only if support of TLS 1.2 is claimed. Send a garbled message from the server after the server has issued the ChangeCipherSpec message and verify that the handshake is not finished successfully and no application data flows. (Note: TLS 1.3 provides for a dummy ChangeCipherSpec message to aid in middlebox compatibility if such an option is enabled in the specific implementation [see Section D.4 in RFC 8446]. If TLS 1.3 middlebox compatibility mode is enabled a ChangeCipherSpec message may appear in packet traces, but it does not influence the protocol. To be clear: for TLS 1.3, this test does not need to be performed.)
Test Steps	• Start the server using the 'acumen-tls' tool and verify that the TOE rejects the connection when a garbled message is sent after the Change CipherSpec message. • Verify the error logs on the device indicating decryption failure. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. Upon receiving garbled data after the ChangeCipherSpec message, the TOE terminates the connection. This meets the test requirements.

6.1.3.61 FCS_TLSC_EXT.1.1 TEST #6C (CPP_ND_V3.0E)

Item	Data

Test Assurance Activity	The evaluator shall perform the following tests: Test 6: The evaluator shall perform the following 'scrambled message tests': iii. [conditional]: Perform this test only if support of TLS 1.3 is claimed. Send a plaintext EncryptedExtensions message from the server and verify that the handshake is not finished successfully and no application data flows. (Note: Under TLS 1.3, the EncryptedExtensions message is the first message to be encrypted with the handshake traffic secret.)
Test Steps	• Use the 'acumen-tls' tool to start the server, send the EncryptedExtensions message in plaintext, and confirm that the TOE terminates the connection. • Verify the error logs on the device for connection failure. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. The TOE terminates the handshake after detecting the plaintext EncryptedExtensions message. No application data flows after the failed handshake. This meets the test requirements.

6.1.3.62 FCS_TLSC_EXT.1.1 TEST #6D (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 6: The evaluator shall perform the following 'scrambled message tests': iv. [conditional]: Perform this test only if support of TLS 1.2 is claimed. Modify at least one byte in the server's nonce in the Server Hello handshake message and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.
Test Steps	• Start the server using the 'acumen-tls' tool and verify that the TOE rejects the connection when a byte is modified in the server's nonce in the Server Hello handshake message. • Verify the error logs showing handshake failure due to a bad signature. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. TOE rejects the connection when the byte is modified in the server's nonce in the Server Hello handshake message. This meets the test requirements.

6.1.3.63 FCS_TLSC_EXT.1.2 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection:

	Test 1 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the CN. Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1.
Note	*Note that the following tests are marked conditional and are applicable under the following conditions:* *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.* *or* *c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.* *Note that for some tests additional conditions apply.* *IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:* • *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.* • *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Test Steps	**Reference identifier as IPv4 address:** • Configure the TOE with the reference identifier name as an IPv4 address. • Create a server certificate with an invalid CN. • Verify that no SAN extension is present in the created certificate. • Establish a connection with the TOE over TLS and verify that the connection fails. • Verify the connection failure logs on the device, indicating that the CN does not match the configured reference identifier.

	• Verify the TLS handshake failure in the packet capture due to the invalid CN. **Reference identifier as FQDN:** • Configure the TOE with the reference identifier name as FQDN. • Create a server certificate with an invalid CN. • Verify that no SAN extension is present in the created certificate. • Establish a connection with the TOE over TLS and verify that the connection fails. • Verify the connection failure logs on the device, indicating that the CN does not match the configured reference identifier. • Verify the TLS handshake failure in the packet capture due to the invalid CN.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when a server certificate containing a CN that does not match the configured reference identifier (IPv4 or FQDN) and does not include the SAN extension is presented. This meets the testing requirements.

6.1.3.64 FCS_TLSC_EXT.1.2 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 2 [conditional]: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, URI). When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the SAN.
Note	*Note that the following tests are marked conditional and are applicable under the following conditions:* *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.* *or*

	c) *For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.* *Note that for some tests additional conditions apply.* *IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:* • *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.* • *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Test Steps	**Reference identifier as IPv4 address:** • Configure the TOE with the reference identifier name as an IPv4 address. • Create a server certificate with a valid CN and an invalid SAN extension. • Establish a connection with the TOE over TLS and verify that the connection fails. • Verify the connection failure logs on the device, indicating that the SAN does not match the configured reference identifier. • Verify the TLS handshake failure in the packet capture due to the invalid SAN. **Reference identifier as FQDN:** • Configure the TOE with the reference identifier name as FQDN. • Create a server certificate with a valid CN and an invalid SAN extension. • Establish a connection with the TOE over TLS and verify that the connection fails. • Verify the connection failure logs on the device, indicating that the SAN does not match the configured reference identifier. • Verify the TLS handshake failure in the packet capture due to the invalid SAN.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when a certificate contains a CN that matches the configured reference identifier (IPv4 or FQDN) but includes a SAN extension that does not match the reference identifier. This meets the testing requirements.

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 3 [conditional]: If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.
Note	*Note that the following tests are marked conditional and are applicable under the following conditions:* *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.* *or* *c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.* *Note that for some tests additional conditions apply.* *IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:* • *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.* • *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Test Steps	**Reference identifier as IPv4 address:** • Configure the TOE with the reference identifier name as an IPv4 address. • Create a server certificate with a valid CN but no SAN extension. • Establish a connection with the TOE over TLS and verify that the connection is established successfully. • Verify that the successful connection is established through packet capture. **Reference identifier as FQDN:** • Configure the TOE with the reference identifier name as an FQDN address.

	• Create a server certificate with a valid CN but no SAN extension. • Establish a connection with the TOE over TLS and verify that the connection is established successfully. • Verify that the successful connection is established through packet capture.
Pass/Fail with Explanation	Pass. The TOE successfully accepts the connection when a server certificate is presented with a CN that matches the reference identifier (IPv4 or FQDN), even if the SAN extension is not included, as the TOE does not mandate its presence. This meets the testing requirements.

6.1.3.66 FCS_TLSC_EXT.1.2 TEST #4 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 4 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, SRV).
Note	*Note that the following tests are marked conditional and are applicable under the following conditions:* *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.* *or* *c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.* *Note that for some tests additional conditions apply.* *IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:* • *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.*

	• *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Test Steps	**Reference identifier as IPv4 address:** • Configure the TOE with the reference identifier name as an IPv4 address. • Create a server certificate with an invalid CN but valid SAN extension. • Establish a connection with the TOE over TLS and verify that the connection is established successfully. • Verify that the successful connection is established through packet capture. **Reference identifier as FQDN:** • Configure the TOE with the reference identifier name as FQDN. • Create a server certificate with an invalid CN but valid SAN extension. • Establish a connection with the TOE over TLS and verify that the connection is established successfully. • Verify that the successful connection is established through packet capture.
Pass/Fail with Explanation	Pass. The TOE successfully accepts the connection when a server certificate is presented with a CN that does not match the reference identifier (IPv4 or FQDN), but includes a SAN extension containing a matching identifier. This meets the testing requirements.

6.1.3.67 FCS_TLSC_EXT.1.2 TEST #5 (1) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID): i. [conditional]: The evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails.
Note	*Note that the following tests are marked conditional and are applicable under the following conditions:* *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.*

	or *c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.* *Note that for some tests additional conditions apply.* *IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:* • *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.* • *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Test Steps	**CN:** • Configure the TOE with the correct reference identifier. • Create the server certificate with a wildcard that is not in the left-most label of CN. • Establish a connection with the TOE over TLS and verify that the connection fails. • Check error logs on the TOE to confirm the certificate validation failure. • Verify the TLS handshake failure in the packet capture. **SAN:** • Configure the TOE with the correct reference identifier. • Create the server certificate with a wildcard that is not in the left-most label of SAN. • Establish a connection with the TOE over TLS and verify that the connection fails. • Check error logs on the TOE to confirm the certificate validation failure. • Verify the TLS handshake failure in the packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when the reference identifier does not match a wildcard in the certificate that is not in the left-most label. This meets the testing requirements.

6.1.3.68 FCS_TLSC_EXT.1.2 TEST #5 (2)(A) (CPP_ND_V3.0E)

Item	Data

Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID): ii. [conditional]: The evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds, if wildcards are supported, or fails if wildcards are not supported. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)
Note	*Note that the following tests are marked conditional and are applicable under the following conditions:* *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.* *or* *c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.* *Note that for some tests additional conditions apply.* *IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:* • *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.* • *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Test Steps	**CN:** • Configure the TOE with the reference identifier using a single left-most label.

	• Create a server certificate with a wildcard in the left-most label in the CN field. • Establish a TLS connection between the TOE and the server and verify that the connection is unsuccessful. • Verify the successful connection through TOE logs. • Verify the successful connection through a packet capture. **SAN:** • Configure the TOE with the reference identifier using a single left-most label. • Create a server certificate with a wildcard in the left-most label in the SAN field. • Establish a TLS connection between the TOE and the server and verify that the connection is unsuccessful. • Verify the successful connection through TOE logs. • Verify the successful connection through a packet capture.
Pass/Fail with Explanation	Pass. The TOE accepts the connection when the reference identifier with a single left-most label matches the wildcard presented in the certificate. This meets the testing requirements.

6.1.3.69 FCS_TLSC_EXT.1.2 TEST #5 (2)(B) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID): ii. [conditional]: The evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)

Note	*Note that the following tests are marked conditional and are applicable under the following conditions:* *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.* *or* *c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.* *Note that for some tests additional conditions apply.* *IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:* • *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.* • *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Test Steps	**CN:** • Configure the TOE with a reference identifier that does not include the left-most label. • Create a server certificate with a wildcard in the left-most label of the CN field. • Establish a TLS connection between the TOE and the server and verify that the connection fails. • Verify the error logs on the TOE indicating the reason for connection failure. • Verify the TLS handshake failure in the packet capture. **SAN:** • Configure the TOE with a reference identifier that does not include the left-most label. • Create a server certificate with a wildcard in the left-most label of the SAN field. • Establish a TLS connection between the TOE and the server and verify that the connection fails.

	• Verify the error logs on the TOE indicating the reason for connection failure. • Verify the TLS handshake failure in the packet capture.
Pass/Fail with Explanation	Pass. When a server certificate containing a wildcard in the left-most label is presented, and the reference identifier is configured without the corresponding left-most label, the TOE correctly rejects the connection. This meets the testing requirements.

6.1.3.70 FCS_TLSC_EXT.1.2 TEST #5 (2)(C) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID): ii. [conditional]: The evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.com) and verify that the connection fails. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)
Note	*Note that the following tests are marked conditional and are applicable under the following conditions:* *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.* *or* *c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.* *Note that for some tests additional conditions apply.*

	IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules: • *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.* • *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Test Steps	**CN:** • Configure the TOE with a reference identifier containing the two left-most labels. • Create a server certificate with a wildcard in the left-most label of the CN field. • Establish a TLS connection between the TOE and the server and verify that the connection fails. • Verify the error logs on the TOE indicating the reason for connection failure. • Verify the TLS handshake failure in the packet capture. **SAN:** • Configure the TOE with a reference identifier containing the two left-most labels. • Create a server certificate with a wildcard in the left-most label of the SAN field. • Establish a TLS connection between the TOE and the server and verify that the connection fails. • Verify the error logs on the TOE indicating the reason for connection failure. • Verify the TLS handshake failure in the packet capture.
Pass/Fail with Explanation	Pass. When configured with a reference identifier containing two left-most labels, the TOE rejects the connection when a server certificate with a wildcard in the left-most label is presented. This meets the testing requirements.

6.1.3.71 FCS_TLSC_EXT.1.2 TEST #6 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection:

	Test 6: Objective: The objective of this test is to ensure the TOE is able to differentiate between IP address identifiers that are not allowed to contain wildcards and other types of identifiers that may contain wildcards. [conditional] If IP address identifiers supported in the SAN or CN, the evaluator shall present a server certificate that contains a CN that matches the reference identifier, except one of the groups has been replaced with a wildcard asterisk (*) (e.g. CN=*.168.0.1 when connecting to 192.168.0.1, CN=2001:0DB8:0000:0000:0008:0800:200C:* when connecting to 2001:0DB8:0000:0000:0008:0800:200C:417A). The certificate shall not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported IP address version (e.g. IPv4, IPv6).
Note	*Note that the following tests are marked conditional and are applicable under the following conditions:* *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.* *or* *c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.* *Note that for some tests additional conditions apply.* *IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:* • *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.* • *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Test Steps	**IPv4:** • Configure the TOE with the correct IPv4 reference identifier. • Create a server certificate without the SAN extension and with a CN that matches the reference identifier, except with one group replaced by an asterisk (*).

	• Establish a TLS connection between the TOE and the server and verify that the connection fails. • Check the TOE logs for certificate validation failure related to CN mismatch. • Verify the TLS handshake failure in the packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when the server certificate is missing the SAN extension and contains a CN that matches the reference identifier IP (IPv4), with one of the groups replaced by an asterisk (*). This meets the testing requirements.

6.1.3.72 FCS_TLSC_EXT.1.2 TEST #7A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 7 [conditional]: If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator shall modify each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier): i. The evaluator shall present a server certificate that does not contain an identifier in the Subject (DN) attribute type(s) that matches the reference identifier. The evaluator shall verify that the connection fails.
Note	*Note that the following tests are marked conditional and are applicable under the following conditions:* *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.* *or* *c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.* *Note that for some tests additional conditions apply.* *IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:*

	• *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.* • *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Pass/Fail with Explanation	Since the TOE uses IPsec—not TLS—for FPT_ITT.1 and for FTP_ITC.1 RFC 6125 is selected, this test is not applicable.

6.1.3.73 FCS_TLSC_EXT.1.2 TEST #7B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 7 [conditional]: If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator shall modify each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier): ii. The evaluator shall present a server certificate that contains a valid identifier as an attribute type other than the expected attribute type (e.g. if the TOE is configured to expect id-at-serialNumber=correct_identifier, the certificate could instead include id-at-name=correct_identifier), and does not contain the SAN extension. The evaluator shall verify that the connection fails. Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass this test.
Note	*Note that the following tests are marked conditional and are applicable under the following conditions:* *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.* *or* *c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.*

	Note that for some tests additional conditions apply. *IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:* • *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.* • *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Pass/Fail with Explanation	Since the TOE uses IPsec—not TLS—for FPT_ITT.1 and for FTP_ITC.1 RFC 6125 is selected, this test is not applicable.

6.1.3.74 FCS_TLSC_EXT.1.2 TEST #7C (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 7 [conditional]: If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator shall modify each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier): iii. The evaluator shall present a server certificate that contains a Subject attribute type that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.
Note	*Note that the following tests are marked conditional and are applicable under the following conditions:* *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.* *or* *c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.*

	Note that for some tests additional conditions apply. *IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:* • *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.* • *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Pass/Fail with Explanation	Since the TOE uses IPsec—not TLS—for FPT_ITT.1 and for FTP_ITC.1 RFC 6125 is selected, this test is not applicable.

6.1.3.75 FCS_TLSC_EXT.1.2 TEST #7D (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: Test 7 [conditional]: If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator shall modify each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier): iv. The evaluator shall confirm that all use of wildcards results in connection failure regardless of whether the wildcards are used in the left or right side of the presented identifier. (Remark: Use of wildcards is not addressed within RFC 5280.)
	Note that the following tests are marked conditional and are applicable under the following conditions: *a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.* *or* *b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable.* *or*

	c) *For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.* *Note that for some tests additional conditions apply.* *IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:* • *IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.* • *IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.*
Pass/Fail with Explanation	Since the TOE uses IPsec—not TLS—for FPT_ITT.1 and for FTP_ITC.1 RFC 6125 is selected, this test is not applicable.

6.1.3.76 FCS_TLSC_EXT.1.3 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows: Test 1: Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds, and a trusted channel can be established.
Test Steps	• Configure the TOE to connect to the TLS server. • Create a complete certificate chain. • Upload the complete certificate chain required for validation to the TOE. • Initiate a connection from the TOE to the TLS server and verify that the connection is successful (i.e., the full certificate chain is properly validated). • Verify the successful connection through a packet capture.
Pass/Fail with Explanation	Pass. The TOE successfully establishes a connection with the TLS server when a complete certificate trust chain is present. This meets the testing requirements.

6.1.3.77 FCS_TLSC_EXT.1.3 TEST #2 (CPP_ND_V3.0E)

Item	Data

Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows: Test 2 [conditional]: If "except with the following administrator override" is selected, the evaluator shall change the presented certificate(s) or modify the operational environment, so that certificate validation fails due to the TSF's inability to determine revocation status. The evaluator shall verify that the certificate is not accepted by the TSF until the Security Administrator authorizes the TSF to establish the connection and this action results in the Trusted Channel being successfully established.
Pass/Fail with Explanation	N/A. The "except with the following administrator override" is not selected.

6.1.3.78 FCS_TLSC_EXT.1.3 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows: Test 3: While performing testing of invalid TLS Client Reference Identifiers, expired X.509 certificates, and invalid X.509 trust chains; the evaluator shall ensure the TSF does not present an administrator override option, with the expcetion of failure to determine revocation status (if selected). Note: This should be a review of behiavor observed while performing other tests.
Test Steps	Failed matching reference Identifier: The requirements of this test case are exercised in FCS_TLSC_EXT.1.2 Test #1 and Test #2. Failed validation of the certificate path: • Remove the Intermediate CA (ICA) from the certificate chain configured on the TOE. • Establish a TLS connection and verify that the connection fails. • Check the TOE logs for an error indicating "CA chain was not found". • Verify the TLS handshake failure in the packet capture. Failed validation of the expiration date: The requirements of this test case are exercised in FIA_X509_EXT.1.1/Rev Test #2. Failed determination of the revocation status: The requirements of this test case are exercised in FIA_X509_EXT.2 Test #1.

Pass/Fail with Explanation	Pass. The failed matching of the reference identifier is validated in FCS_TLSC_EXT.1.2 Test #1 and Test #2. The TOE rejects the connection when an incomplete certificate trust chain is present. Rejection of expired certificates is validated in FIA_X509_EXT.1.1/Rev Test #2. The behavior for failed determination of revocation status is confirmed in FIA_X509_EXT.2 Test #1. This meets the testing requirements.

6.1.3.79 FCS_TLSC_EXT.1.4 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1 [conditional]: If “not present the Supported Groups Extension” is selected, the evaluator shall examine the Client Hello message and verify it does not contain the Supported Groups extension.
Pass/Fail with Explanation	N/A. This is not applicable as the “not present the Supported Groups Extension” is not selected in ST.

6.1.3.80 FCS_TLSC_EXT.1.4 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2 [conditional]: If “present the Supported Groups Extension” is selected, the evaluator shall configure the server to perform ECDHE or DHE (as applicable) key exchange using each of the TOE’s supported groups. The evaluator shall verify that the connection succeeds. This test shall be repeated for each type of key exchange message/extension supported (i.e. Key Share extension for TLS 1.3 and Server Key Exchange Message for TLS 1.2).
Test Steps	TLS1.2 • Establish a connection with the TOE over TLS using the curve secp384r1 and verify that it is successful. • Verify through packet capture that the connection is established using the specified curve. • Establish a connection with the TOE over TLS using the curve secp521r1 and verify that it is successful. • Verify through packet capture that the connection is established using the specified curve.

	TLS1.3 • Establish a connection with the TOE over TLS using the curve secp384r1 and verify that it is successful. • Verify through packet capture that the connection is established using the specified curve. • Establish a connection with the TOE over TLS using the curve secp521r1 and verify that it is successful. • Verify through packet capture that the connection is established using the specified curve.
Pass/Fail with Explanation	Pass. The TOE successfully establishes a TLS connection with the server for all supported groups/curves, using the correct key exchange method for each protocol version. This meets the testing requirements.

6.1.3.81 FCS_TLSC_EXT.1.4 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 3 [conditional]: If secp curves are selected, the evaluator shall configure the server to perform an ECDHE key exchange in the TLS connection using a non-supported curve and shall verify that the connection fails and no application data flows. The non-supported curve shall be as similar to the selected curve(s) as possible (i.e. a non-selected curve when not all curves are selected or P-224). This test shall be repeated for each type of key exchange message/extension supported (i.e. Key Share extension for TLS 1.3 and Server Key Exchange Message for TLS 1.2).
Test Steps	TLS1.2: • Start the server using the 'acumen-tls' tool and try to establish the connection using non-supported curve P-224 and verify that it is unsuccessful. • Verify error logs generated on the TOE. • Verify through packet capture that the connection is unsuccessful and there is no application data flow. TLS1.3: • Start the server using the 'acumen-tls' tool and try to establish the connection using non-supported curve P-224 and verify that it is unsuccessful. • Verify error logs generated on the TOE. • Verify through packet capture that the connection is unsuccessful and there is no application data flow.

Pass/Fail with Explanation	Pass. The TOE rejects the connection when the server presents unsupported elliptic curves for both TLS 1.2 and TLS 1.3. This meets the testing requirements.

6.1.3.82 FCS_TLSC_EXT.1.4 TEST #4A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 4a [conditional, for TLS 1.3 only]: If ffdhe curves are selected, the evaluator shall configure the server to perform a DHE key exchange in the TLS connection using a non-supported group and shall verify that the connection fails and no application data flows. The non-supported group shall be as similar to the selected group(s) as possible (i.e. a non-selected group when not all groups are selected or undefined Codepoint 0x0105 (ffdhe8192 + 1)).
Pass/Fail with Explanation	The ffdhe curves are not selected in ST; hence, this activity is not applicable.

6.1.3.83 FCS_TLSC_EXT.1.4 TEST #4B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 4b [conditional, for TLS 1.2 only]: If ffdhe curves are selected, the evaluator shall configure the server to return DHE parameters in the Server Key Exchange in the TLS connection that do not meet the construction for any claimed ffdhe group. The evaluator shall verify that the connection fails and no application data flows. If the TOE client supports any server-returned DHE parameter set, then this test is not applicable.
Pass/Fail with Explanation	The ffdhe curves are not selected in ST; hence, this activity is not applicable.

6.1.3.84 FCS_TLSC_EXT.1.5 TEST #1A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1 [conditional]: The evaluator shall perform the following tests if “present the signature_algorithms extension” is selected:

	i. The evaluator shall examine the Client Hello message and verify it contains the signature_algorithms extension and the SignatureSchemes match the SignatureSchemes specified in the requirement.
Test Steps	• Establish a connection with the TOE over TLS and verify that it is successful. • Verify through packet capture that the Client Hello message contains the signature_algorithms extension and the SignatureSchemes match the SignatureSchemes specified in the requirement.
Pass/Fail with Explanation	Pass. The Client Hello message contains the signature_algorithms extension, and the SignatureSchemes matchs the SignatureSchemes specified in the requirement. This meets the testing requirements.

6.1.3.85 FCS_TLSC_EXT.1.5 TEST #1B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1 [conditional]: The evaluator shall perform the following tests if "present the signature_algorithms extension" is selected: ii. The evaluator shall establish a TLS connection using each of the SignatureSchemes specified by the requirement and observes the session is successfully completed. The evaluator shall ensure the test server sends a leaf Certificate that has a public key algorithm that is consistent with the SignatureScheme being tested. **For TLS 1.2 and if the ciphersuite is DHE or ECDHE**, the evaluator shall ensure that the server sends Server Key Exchange messages consistent with the SignatureScheme being tested. **For TLS 1.3**, the evaluator shall ensure that the server sends Certificate Verify messages consistent with the SignatureScheme being tested.
Test Steps	TLS1.2: • Establish a connection with the TOE over TLS using the algorithm ecdsa_secp384r1 with sha384 and verify that it is successful. • Verify through packet capture that the connection is established using the algorithm ecdsa_secp384r1 with sha384. • Establish a connection with the TOE over TLS using the algorithms ecdsa_secp521r1 with sha512 and verify that it is successful. • Verify through packet capture that the connection is established using the algorithm ecdsa_secp521r1 with sha512. TLS1.3: • Establish a connection with the TOE over TLS using the algorithm ecdsa_secp384r1 with sha384 and verify that it is successful. • Verify through packet capture that the connection is established using the algorithm ecdsa_secp384r1 with sha384.

	• Establish a connection with the TOE over TLS using the algorithms ecdsa_secp521r1 with sha512 and verify that it is successful. • Verify through packet capture that the connection is established using the algorithm ecdsa_secp521r1 with sha512.
Pass/Fail with Explanation	Pass. The TLS connection can be successfully established using each specified SignatureScheme. This meets the testing requirements.

6.1.3.86 FCS_TLSC_EXT.1.5 TEST #2A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2 [conditional]: The evaluator shall perform the following tests if “present the signature_algorithms_cert extension” is selected: i. The evaluator shall examine the Client Hello message and verify it contains the signature_algorithms_cert extension and the SignatureSchemes match the SignatureSchemes specified in the requirement.
Pass/Fail with Explanation	The selection “present the signature_algorithms_cert extension” is not selected in ST; hence, this activity is not applicable.

6.1.3.87 FCS_TLSC_EXT.1.5 TEST #2B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2 [conditional]: The evaluator shall perform the following tests if “present the signature_algorithms_cert extension” is selected: ii. The evaluator shall establish a TLS connection using a certificate chain using each of the SignatureSchemes specified by the requirement. The evaluator shall ensure the signatures used in the certificate chain are consistent with the SignatureScheme being tested.
Pass/Fail with Explanation	The selection “present the signature_algorithms_cert extension” is not selected in ST; hence, this activity is not applicable.

6.1.3.88 FCS_TLSC_EXT.1.6 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	[conditional]: If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall establish a TLS connection using one of the possible configurations of the list of supported ciphersuites. The evaluator shall then change

	the configuration and repeat the test. The evaluator shall verify that the behavior of the TOE has changed according to the modification of the list of ciphers. This test shall be repeated for all supported TLS versions. If the TSF does not provide the ability of configuring the list of supported ciphersuites, this test shall be omitted.
Pass/Fail with Explanation	Pass. The ciphers TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 and TLS_AES_256_GCM_SHA384 were tested in FCS_TLSC_EXT.1.1 Test #1.

6.1.3.89 FCS_TLSC_EXT.1.7 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall establish a TLS connection with a server and observe that the early data extension and the post-handshake client authentication extension according to RFC 8446 Section 4.2 are not advertised in the Client Hello Message. This test shall be executed for all TLS versions supported by the TOE.
Test Steps	TLS 1.2: • Establish a connection with the TOE over TLS. • Verify through packet capture that early data extension and post-handshake client authentication extension are not advertised in the Client Hello message. TLS 1.3: • Establish a connection with the TOE over TLS. • Verify through packet capture that early data extension and post-handshake client authentication extension are not advertised in the Client Hello message.
Pass/Fail with Explanation	Pass. The early data extension and post-handshake client authentication extension are not advertised in the Client Hello message for all TLS versions supported by the TOE. This meets the testing requirements.

6.1.3.90 FCS_TLSC_EXT.1.8 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	No test assurance activities have been identified.
Pass/Fail with Explanation	No test assurance activities have been identified.

6.1.3.91 FCS_TLSC_EXT.1.9 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1 [conditional]: If "support TLS 1.2 secure renegotiation…" is selected, the evaluator shall use a network packet analyzer/sniffer to capture a TLS 1.2 handshake between the two TLS endpoints. The evaluator shall verify that either the “renegotiation_info” field or the SCSV ciphersuite is included in the ClientHello message during the initial handshake.
Pass/Fail with Explanation	The selection “support TLS 1.2 secure renegotiation…” is not selected in ST; hence, this activity is not applicable.

6.1.3.92 FCS_TLSC_EXT.1.9 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2 [conditional]: If "support TLS 1.2 secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and verify the TOE TLS Client’s handling of ServerHello messages received during the initial handshake that include the “renegotiation_info” extension. The evaluator shall modify the length portion of this field in the ServerHello message to be non-zero and verify that the TOE TLS client sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.
Pass/Fail with Explanation	The selection “support TLS 1.2 secure renegotiation…” is not selected in ST; hence, this activity is not applicable.

6.1.3.93 FCS_TLSC_EXT.1.9 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 3 [conditional]: If "support TLS 1.2 secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and verify that ServerHello messages received during secure renegotiation contain the “renegotiation_info” extension. The evaluator shall modify either the “client_verify_data” or “server_verify_data” value and verify that the TOE TLS client terminates the connection.
Pass/Fail with Explanation	The selection “support TLS 1.2 secure renegotiation…” is not selected in ST; hence, this activity is not applicable.

6.1.3.94 FCS_TLSC_EXT.1.9 TEST #4A (CPP_ND_V3.0E) **[TD0899]**

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 4a [conditional, if the TOE supports TLS 1.3]: The evaluator shall initiate a TLS session between the TSF and a test TLS 1.3 server that completes a compliant TLS 1.3 handshake, followed by a hello request message. The evaluator shall observe that the TSF completes the initial TLS 1.3 handshake successfully, but terminates the session on receiving the hello request message. It is preferred that the TSF sends a fatal error alert message (e.g., unexpected message) in response to this, but it is acceptable that the TSF terminates the connection silently (i.e., without sending a fatal error alert). **TD0899 has been applied**.
Test Steps	• Using the acumen-tls tool, establish a successful TLS 1.3 handshake with the TOE and then send an unexpected HelloRequest message. Verify that the TOE terminates the connection. • Verify through packet capture that the TLS session is closed after receiving the HelloRequest message.
Pass/Fail with Explanation	Pass. The TOE terminates the TLS 1.3 session when it receives a HelloRequest message, which is not part of the TLS 1.3 protocol. This meets the testing requirements.

6.1.3.95 FCS_TLSC_EXT.1.9 TEST #4B (CPP_ND_V3.0E) **[TD0899]**

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 4b [conditional, if the TOE supports TLS 1.2 and rejects TLS 1.2 renegotiation attempts]: The evaluator shall initiate a TLS session between the so-configured TSF and a test TLS 1.2 server that is configured to perform a compliant handshake, followed by a hello request. The evaluator shall confirm that the TSF completes the initial handshake successfully but does not initiate renegotiation after receiving the hello request. **TD0899 has been applied**.

Test Steps	• Using the acumen-tls tool, establish a successful TLS 1.2 handshake with the TOE, then send a HelloRequest message from the server. Verify that the TOE does not initiate renegotiation in response. • Verify through packet capture that TOE completes the initial handshake successfully but does not initiate renegotiation after receiving the hello request.
Pass/Fail with Explanation	Pass. The TOE successfully establishes the initial handshake and does not initiate renegotiation after receiving the hello request. This meets the testing requirements.

6.1.3.96 FCS_TLSS_EXT.1.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
Test Steps	TLSv1.2: • Establish a connection with the TOE over TLS using the ciphersuite TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384. • Verify through packet capture that the required cipher is used in a successful TLS connection. TLSv1.3: • Establish a connection with the TOE over TLS using the ciphersuite TLS_AES_256_GCM_SHA384. • Verify through packet capture that the required cipher is used in a successful TLS connection.
Pass/Fail with Explanation	Pass. TOE successfully negotiates each of the claimed ciphersuites. This meets the testing requirements.

6.1.3.97 FCS_TLSS_EXT.1.1 TEST #2A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: The evaluator shall perform the following tests: i. The evaluator shall send a Client Hello to the server with a list of ciphersuites that does not contain any of the ciphersuites in the server's ST and verify that the server denies the connection.
Test Steps	• Using the acumen-tls tool as a client, attempt to establish a TLS connection to the TOE using an unsupported cipher suite: TLS_RSA_WITH_3DES_EDE_CBC_SHA. The connection attempt should fail. • Verify the failure logs on the TOE showing failure due to no shared cipher. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects TLS connections when unsupported ciphersuites are offered. This meets the testing requirement.

6.1.3.98 FCS_TLSS_EXT.1.1 TEST #2B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: The evaluator shall perform the following tests: ii. [conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the server denies the connection.
Test Steps	• Using the acumen-tls tool as a client, attempt to establish a TLS connection to the TOE by including the unsupported cipher suite TLS_NULL_WITH_NULL_NULL in the ClientHello message. The connection attempt should fail. • Verify that the TOE logs show a failure due to no shared cipher. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects TLS connections with the unsupported TLS_NULL_WITH_NULL_NULL ciphersuites. This meets the testing requirement.

6.1.3.99 FCS_TLSS_EXT.1.1 TEST #3A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 3: The evaluator shall perform the following modifications to the traffic:

	i. [conditional]: Perform this test only if support of TLS 1.2 is claimed. Modify a byte in the Client Finished handshake message, and verify that the server rejects the connection and does not send any application data.
Test Steps	• Run the 'acumen-tls' tool as a client with a modified client finished message and wait for the connection, the connection should fail. • Verify the failure logs on the device, ensuring they indicate the failure occurred due to a decryption failure. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. The TOE correctly rejects the connection upon receiving the modified Client Finished message. This meets the testing requirements.

6.1.3.100 FCS_TLSS_EXT.1.1 TEST #3B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 3: The evaluator shall perform the following modifications to the traffic: ii. (The intent of this test is to ensure that the server's TLS implementation immediately makes use of the key exchange and authentication algorithms to: a) Correctly encrypt TLS Finished message and b) Encrypt every TLS message after session keys are negotiated.) [conditional]: Perform this test only if support of TLS 1.2 is claimed. The evaluator shall use one of the claimed ciphersuites to complete a successful handshake and observe transmission of properly encrypted application data. The evaluator shall verify that no Alert with alert level Fatal (2) messages were sent. The evaluator shall verify that the Finished message (Content type hexadecimal 16 and handshake message type hexadecimal 14) is sent immediately after the server's ChangeCipherSpec (Content type hexadecimal 14) message. The evaluator shall examine the Finished message (encrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 11 22 33 44 55…) and confirm that it does not contain unencrypted data (unencrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 14 00 00 0c…), by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14 for at least one of three test messages. There is a chance that an encrypted Finished message contains a hexadecimal value of '14' at the position where a plaintext Finished message would contain the message type code '14'. If the observed Finished message contains a hexadecimal value of '14' at the position where the plaintext Finished message would contain the message type code, the test shall

	be repeated three times in total. In case the value of '14' can be observed in all three tests it can be assumed that the Finished message has indeed been sent in plaintext and the test has to be regarded as 'failed'. Otherwise it has to be assumed that the observation of the value '14' has been due to chance and that the Finished message has indeed been sent encrypted. In that latter case the test shall be regarded as 'passed'.
Test Steps	• Initiate a connection to the TOE with acumen-tls tool as a client. • Verify that no Alert with alert level Fatal (2) messages were sent. • Verify that the server's ChangeCipherSpec message (hex 14) is immediately followed by the Finished message (TLS Content Type hex 16, Handshake Type hex 14). • Examine the Finished message and confirm that it does not contain unencrypted data by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14.
Pass/Fail with Explanation	Pass. The Finished message (Content Type hex 16) is sent immediately after the ChangeCipherSpec message (hex 14). The first byte of the encrypted Finished message is not hex 14, indicating that the message is properly encrypted. This meets the testing requirements.

6.1.3.101 FCS_TLSS_EXT.1.1 TEST #3C (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 3: The evaluator shall perform the following modifications to the traffic: iii. [conditional]: Perform this test only if support of TLS 1.3 is claimed. The evaluator shall use a client to send a Client Hello message containing a single curve in the Supported Groups extension. The curve that is selected to be presented in this extension should not be supported by the TOE. The evaluator shall verify that the TOE disconnects after receiving the Client Hello message.
Test Steps	• Run the 'acumen-tls' tool as a client with only an unsupported elliptic curve in the Supported Groups extension and verify that the TOE rejects the handshake. • Confirm that device logs indicate failure due to "no suitable key share." • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. The TOE correctly rejects the TLS 1.3 handshake when an unsupported curve is used. This meets the testing requirements.

6.1.3.102 FCS_TLSS_EXT.1.1 TEST #3D (CPP_ND_V3.0E)

Item	Data

Test Assurance Activity	The evaluator shall perform the following tests: Test 3: The evaluator shall perform the following modifications to the traffic: iv. [conditional]: Perform this test only if support of TLS 1.3 is claimed. The evaluator shall use a client to send a Client Hello message containing multiple curves in the Supported Groups extension. These curves should be chosen such that only one of these curves is supported by the TOE. The evaluator shall verify that the TOE responds with a Hello Retry Request message selecting the supported curve. This shall be reflected in the Key Share extension of the Hello Retry Request message.
Test Steps	• Run the acumen-tls tool to send a TLS Client Hello message containing multiple curves in the Supported Groups extension, ensuring that only one of these curves is supported by the TOE. • Verify through packet capture that the TOE responds with a Hello Retry Request message and ensure that a successful connection is established with a supported curve in the Key Share extension.
Pass/Fail with Explanation	Pass. The TOE correctly identifies the single supported curve among the multiple curves provided in the Client Hello message. The TOE responds with a Hello Retry Request message, selecting the supported curve and reflecting it in the Key Share extension. This meets the testing requirements.

6.1.3.103 FCS_TLSS_EXT.1.1 TEST #4 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 4: The evaluator shall attempt to establish a TLS/SSL connection using each of the supported TLS/SSL versions (i.e., TLS 1.3, TLS 1.2, TLS 1.1, TLS 1.0, SSL 3.0, SSL 2.0). The client shall be configured so it only supports the version being tested. The evaluator shall verify that the versions specified in FCS_TLSS_EXT.1.1 are successfully established and all other versions not successfully established. If the TOE attempts to downgrade the version, it is acceptable for the test client to terminate the connection; however, the version selected by the TOE shall always be a version specified in FCS_TLSS_EXT.1.1.
Test Steps	• Using the acumen-tls tool as a client, attempt an SSL v2.0 connection to the TOE. • Verify the failed connection with the logs. • Verify that the SSL v2.0 connection is denied using a packet capture. • Using the acumen-tls tool as a client, attempt an SSL v3.0 connection to the TOE. • Verify the failed connection with the logs. • Verify that the SSL v3.0 connection is denied using a packet capture. • Using the acumen-tls tool as a client, attempt a TLS v1.0 connection to the TOE.

	• Verify the failed connection with the logs. • Verify that the TLS v1.0 connection is denied using a packet capture. • Using the acumen-tls tool as a client, attempt a TLS v1.1 connection to the TOE. • Verify the failed connection with the logs. • Verify that the TLS v1.1 connection is denied using a packet capture. • Attempt a TLS v1.2 connection to the TOE. • Verify that the TLS v1.2 connection is successful using a packet capture. • Attempt a TLS v1.3 connection to the TOE. • Verify that the TLS v1.3 connection is successful using a packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects all SSLv2, SSLv3, TLS v1.0, and TLS v1.1 connection attempts and accepts the connection when TLS 1.2 or TLS 1.3 is used. This meets the testing requirement.

6.1.3.104 FCS_TLSS_EXT.1.2 & 1.3 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1 [conditional]: If ECDHE ciphersuites/group are supported: The evaluator shall repeat this test for each supported elliptic curve. The evaluator shall attempt a connection using a supported ECDHE ciphersuite (TLS 1.2) or group (TLS 1.3) and a single supported elliptic curve specified in the supported groups extension. The Evaluator shall verify (through a packet capture or instrumented client) that the TOE selects the same curve in the Server Key Exchange (TLS 1.2) or Server Hello (key_share, for TLS 1.3) message and successfully establishes the connection. **For TLS 1.2**, the evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single unsupported elliptic curve (e.g., secp192r1 (0x13)) specified in RFC 4492, Section 5.1.1. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established. **For TLS 1.3**, the evaluator shall attempt a connection using a supported ciphersuite and a single unsupported group. Both the key_share and supported_groups extensions must be set to the same unsupported group. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.
Test Steps	**Part 1: Positive Tests with Supported Curves** **For TLS 1.2:**

	• Connect to the TOE over TLS using the secp384r1 curve and verify that the connection is successfully established. • Verify through packet capture that the connection is successful and the selected curve is secp384r1 in the Server Key Exchange message. • Connect to the TOE over TLS using the secp521r1 curve and verify that the connection is successfully established. • Verify through packet capture that the connection is successful and the selected curve is secp521r1 in the Server Key Exchange message. **For TLS 1.3:** • Connect to the TOE over TLS using the secp384r1 curve and verify that the connection is successfully established. • Verify through packet capture that the connection is successful and the curve secp384r1 appears in the key_share extension of the Server Hello message. • Connect to the TOE over TLS using the secp521r1 curve and verify that the connection is successfully established. • Verify through packet capture that the connection is successful and the curve secp521r1 appears in the key_share extension of the Server Hello message. **Part 2: Negative Tests with Unsupported Curves.** **For TLS 1.2:** • Use the Acumen-TLS tool to attempt a connection using a supported ECDHE cipher suite and an unsupported curve. verify that the connection fails. • Verify that the TOE logs show failure due to no shared cipher. • Verify the TLS handshake failure by analyzing the packet capture. **For TLS 1.3:** • Use the Acumen-TLS tool to attempt a connection using a supported ECDHE cipher suite and an unsupported curve. verify that the connection fails. • Verify that TOE logs indicate "no suitable key share“. • Verify the TLS handshake failure by analyzing the packet capture.
Pass/Fail with Explanation	Pass. The TOE correctly selects the supported elliptic curve when offered and establishes the connection. When an unsupported curve is presented, the TOE correctly rejects the connection, and no Server Hello is sent. This meets the testing requirements.

6.1.3.105 FCS_TLSS_EXT.1.2 & 1.3 TEST #2A (CPP_ND_V3.0E)**[TD0973]**

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2a [conditional]: If the TOE supports TLS 1.2 and DHE ciphersuites, the evaluator shall repeat the following test for each supported parameter size. If any configuration is necessary, the evaluator shall configure the TOE to use each supported Diffie-Hellman parameter size. The evaluator shall attempt a connection using a supported DHE ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Exchange Message where p Length is consistent with the ones configured Diffie-Hellman parameter size(s).
Pass/Fail with Explanation	NA. The TOE does not support DHE ciphersuites; hence, this activity is not applicable.

6.1.3.106 FCS_TLSS_EXT.1.2 & 1.3 TEST #2B (CPP_ND_V3.0E)**[TD0973]**

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2b: [conditional] If the TOE supports TLS 1.3 and 'ffdhe' parameters are selected, the evaluator shall repeat the following test for each supported FFDHE parameter size. If any configuration is necessary, the evaluator shall configure the TOE to use each supported Finite Field Diffie-Hellman parameter size. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Share Extension Message where the KeyShareServerHello structure contains a KeyShareEntry structure with an opaque key_exchange value whose Length is consistent with the configured Finite Field Diffie-Hellman parameter size(s).
Pass/Fail with Explanation	NA. The TOE does not support DHE ciphersuites; hence, this activity is not applicable.

6.1.3.107 FCS_TLSS_EXT.1.2 & 1.3 TEST #3 (CPP_ND_V3.0E)

Item	Data

Test Assurance Activity	The evaluator shall perform the following tests: Test 3 [conditional]: If RSA key establishment ciphersuites are supported, the evaluator shall repeat this test for each RSA key establishment key size. If any configuration is necessary, the evaluator shall configure the TOE to perform RSA key establishment using a supported key size (e.g. by loading a certificate with the appropriate key size). The evaluator shall attempt a connection using a supported RSA key establishment ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a certificate whose modulus is consistent with the configured RSA key size.
Pass/Fail with Explanation	NA. The TOE does not support RSA key establishment ciphersuites; hence, this activity is not applicable.

6.1.3.108 FCS_TLSS_EXT.1.4 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	*Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption).* The evaluator shall perform the following tests: Test 1 [conditional]: If the TOE does not support session resumption based on session IDs according to RFC 5246 (TLS 1.2) or session tickets according to RFC 5077 (TLS 1.2) or session resumption according to RFC 8446 (TLS 1.3), the evaluator shall perform the following test: i. For all supported TLS versions the client shall send a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket. A non-zero length session identifier for TLS 1.3 would result in testing compatibility mode which is not the objective of this test. For TLS 1.3, the evaluator shall ensure that a 'psk_key_exchange_modes' extension is included in the Client Hello. ii. The client verifies the server does not send a NewSessionTicket handshake message (at any point in the handshake). iii. The client verifies the Server Hello message contains a zerolength session identifier. For TLS 1.2 the client could alternatively pass the following steps (not applicable for TLS 1.3): Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID. iv. The client completes the TLS handshake and captures the SessionID from the ServerHello. v. The client sends a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d).

	vi. The client verifies the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data. Remark: If multiple contexts are supported for session resumption, the session ID or session ticket may be obtained in one context for resumption in another context. It is possible that one or more contexts may only permit the construction of sessions to be reused in other contexts but not actually permit resumption themselves. For contexts which do not permit resumption, the evaluator is required to verify this behaviour subject to the description provided in the TSS. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.
Test Steps	TLS1.2: • Establish a connection with the TOE over TLS. • Verify through packet capture that the client sends a client hello message with a zero-length session identifier and a session ticket extension containing a zero-length ticket. • Verify through packet capture that the server does not send a new session ticket handshake message at any point during the handshake and that the server hello message contains a zero-length session identifier. TLS1.3 • Establish a connection with the TOE over TLS. • Verify through packet capture that the 'psk_key_exchange_modes' extension is included in the Client Hello. • Verify through packet capture that the server does not send a new session ticket handshake message at any point during the handshake.
Pass/Fail with Explanation	Pass. This TOE does not support session resumption. For TLS 1.2, the Client Hello message includes a zero-length session identifier and a SessionTicket extension containing a zero-length ticket. For TLS 1.3, the Client Hello message includes the psk_key_exchange_modes extension. For both versions, the server does not send a NewSessionTicket handshake message at any point during the handshake. This meets the testing requirement.

6.1.3.109 FCS_TLSS_EXT.1.4 TEST #2A (CPP_ND_V3.0E)

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

	Test 2 [conditional]: If the TOE supports session resumption using session IDs according to RFC 5246 (TLS 1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): i. The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246). When the session is resumed, the evaluator shall verify on the TLS Client used for performing this test, that the TOE (TLS Server) has not advertised support for the early data extension.
Pass/Fail with Explanation	NA. The TOE does not support session resumption using session IDs according to RFC 5246 (TLS 1.2); hence, this activity is not applicable.

6.1.3.110 FCS_TLSS_EXT.1.4 TEST #2B (CPP_ND_V3.0E)

Item	**Data**
Test Assurance Activity	*Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption).* Test 2 [conditional]: If the TOE supports session resumption using session IDs according to RFC 5246 (TLS 1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS): ii. The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data. Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ID may be obtained in one context for resumption in

	another context. There is no requirement that the session ID be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ID constructed in another context must be tested. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.
Pass/Fail with Explanation	NA. The TOE does not support session resumption using session IDs according to RFC 5246 (TLS 1.2); hence, this activity is not applicable.

6.1.3.111 FCS_TLSS_EXT.1.4 TEST #3A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	*Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption).* Test 3 [conditional]: If the TOE supports session tickets according to RFC 5077 (supported only by TLS 1.2), the evaluator shall carry out the following steps: i. The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE responds with an abbreviated handshake described in Section 3.1 of RFC 5077 and illustrated with an example in figure 2. Of particular note: if the server successfully verifies the client's ticket, then it may renew the ticket by including a NewSessionTicket handshake message after the ServerHello in the abbreviated handshake (which is shown in figure 2). This is not required, however as further clarified in Section 3.3 of RFC 5077. When the session is resumed, the evaluator shall verify on the TLS Client used for performing this test, that the TOE (TLS Server) has not advertised support for the early data extension.
Pass/Fail with Explanation	NA. The TOE does not support session tickets according to RFC 5077 (TLS 1.2); hence, this activity is not applicable."

6.1.3.112 FCS_TLSS_EXT.1.4 TEST #3B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	*Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption).* Test 3 [conditional]: If the TOE supports session tickets according to RFC 5077 (supported only by TLS 1.2), the evaluator shall carry out the following steps:

	ii. The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data. Remark: If multiple contexts are supported for session resumption, for each of the above test cases, the session ticket may be obtained in one context for resumption in another context. There is no requirement that the session ticket be obtained and replayed within the same context subject to the description provided in the TSS. All contexts that can reuse a session ticket constructed in another context must be tested. It is not mandated that the session establishment and session resumption share context. For example, it is acceptable for a control channel to establish and application channel to resume the session.
Pass/Fail with Explanation	NA. The TOE does not support session tickets according to RFC 5077 (TLS 1.2); hence, this activity is not applicable.

6.1.3.113 FCS_TLSS_EXT.1.4 TEST #4A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	*Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption).* Test 4 [conditional]: If the TOE supports session resumption according to RFC 8446 (supported only by TLS 1.3), the evaluator shall carry out the following steps: i. The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the pre-shared key in the ClientHello. The evaluator shall confirm that the TOE responds similarly to figure 3 of RFC 8446 after successfully reusing the pre-shared-key to resume the session. Specifically, the server must not send back a Certificate message if the session is correctly resumed. When the session is resumed, the evaluator shall verify on the TLS Client used for performing this test, that the TOE (TLS Server) has not advertised support for the early data extension.
Pass/Fail with Explanation	NA. The TOE does not support session resumption according to RFC 8446; hence, this activity is not applicable.

6.1.3.114 FCS_TLSS_EXT.1.4 TEST #4B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	*Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption).* Test 4 [conditional]: If the TOE supports session resumption according to RFC 8446 (supported only by TLS 1.3), the evaluator shall carry out the following steps: ii. The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then modify the pre-shared key and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake, or (2) terminates the connection in some way that prevents the flow of application data.
Pass/Fail with Explanation	NA. The TOE does not support session resumption according to RFC 8446; hence, this activity is not applicable.

6.1.3.115 FCS_TLSS_EXT.1.4 TEST #4C (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	*Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption).* Test 4 [conditional]: If the TOE supports session resumption according to RFC 8446 (supported only by TLS 1.3), the evaluator shall carry out the following steps: iii. The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then force the non-TOE client to attempt to establish a new connection using the previous session ticket material as a pre-shared key, but set psk_key_exchange_modes with a value of psk_ke in the Client Hello message and omit the psk_ke_dhe. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake, or (2) terminates the connection in some way that prevents the flow of application data.
Pass/Fail with Explanation	NA. The TOE does not support session resumption according to RFC 8446; hence, this activity is not applicable.

6.1.3.116 FCS_TLSS_EXT.1.5 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Test 1[conditional]: If the TSF provides the ability of configuring the list of supported ciphersuites, the evaluator shall establish a TLS connection using one of

	the possible configurations of the list of supported ciphersuites. The evaluator shall then change the configuration and repeat the test. The evaluator shall verify that the behavior of the TOE has changed according to the modification of the list of ciphers. This test shall be repeated for all supported TLS versions. If the TSF does not provide the ability of configuring the list of supported ciphersuites, this test shall be omitted.
Pass/Fail with Explanation	NA. The TSF does not provide the ability to configure the list of supported ciphersuites; hence, this activity is not applicable.

6.1.3.117 FCS_TLSS_EXT.1.6 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	According to RFC 8446 Section 4.2.10, a PSK is required to use the early data extension. As NDcPP only allows the use of PSK in conjunction with session resumption, a NDcPP conformant TOE which acts as TLS Server cannot use the early data extension if session resumption is not supported. For TOEs that do not support session resumption, execution of test FCS_TLSS_EXT.1.4 Test 1 is regarded as sufficient that the TOE does not support the early data extension. For TOEs that support session resumption, FCS_TLSS_EXT.1.4 Test 2(i), 3(i) or 4(i) (depending on the supported TLS versions and the way session resumption is implemented) ensure that the TOE does not support the early data extension.
Pass/Fail with Explanation	Pass. The test was performed under FCS_TLSS_EXT.1.4 Test #1, demonstrating that the TOE does not support session resumption for both TLS 1.2 and TLS 1.3.

6.1.3.118 FCS_TLSS_EXT.1.7 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	No test assurance activities have been identified.
Pass/Fail with Explanation	No test assurance activities have been identified.

6.1.3.119 FCS_TLSS_EXT.1.8 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1 [conditional]: If "support secure renegotiation…" is selected, the evaluator shall use a network packet analyzer/sniffer to capture a TLS 1.2 handshake between the two TLS endpoints. The evaluator shall verify that the “renegotiation_info” extension is included in the ServerHello message.
Pass/Fail with Explanation	NA. The ST does not select "support secure renegotiation…", hence this activity is not appliable.

6.1.3.120 FCS_TLSS_EXT.1.8 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2 [conditional]: If "support secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and modify the length portion of the field in the ClientHello message in the initial handshake to be non-zero. The evaluator shall verify that the TOE TLS server sends a failure and terminates the connection. The evaluator shall verify that a properly formatted field results in a successful TLS connection.
Pass/Fail with Explanation	NA. The ST does not select "support secure renegotiation…", hence this activity is not appliable.

6.1.3.121 FCS_TLSS_EXT.1.8 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 3 [conditional]: If "support secure renegotiation…" is selected, the evaluator shall perform a TLS 1.2 handshake and modify the "client_verify_data" or "server_verify_data" value in the ClientHello message received during secure renegotiation. The evaluator shall verify that the TOE TLS server terminates the connection.
Pass/Fail with Explanation	The "support secure renegotiation…" is not selected in the ST; hence, this activity is not applicable.

6.1.3.122 FCS_TLSS_EXT.1.8 TEST #4A (CPP_ND_V3.0E) **[TD0899]**

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 4a [conditional, if the TOE supports TLS 1.3]: The evaluator shall follow the operational guidance as necessary to configure the TSF to negotiate TLS 1.3. The evaluator shall initiate a valid initial TLS 1.3 session, send a valid client hello on the non-renegotiable TLS channel, and observe that the TSF terminates the session. Note: It is preferred that the TSF sends a fatal error alert message (e.g., unexpected message) in response to this, but it is acceptable that the TSF terminates the connection silently (i.e., without sending a fatal error alert). **TD0899 has been applied.**

Test Steps	• Using the acumen-tls tool, send a valid ClientHello message to start a session using TLS version 1.3 and monitor the connection to ensure the TOE terminates the session after receiving the renegotiation request. • Verify through packet capture that the TOE terminates the session after receiving the renegotiation request.
Pass/Fail with Explanation	Pass. The TOE correctly negotiates the specified TLS version and terminates the session when renegotiation is attempted. This meets the testing requirements.

6.1.3.123 FCS_TLSS_EXT.1.8 TEST #4B (CPP_ND_V3.0E) **[TD0899]**

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 4b [conditional, if the TOE supports TLS 1.2 and rejects TLS 1.2 renegotiation attempts]: The evaluator shall follow the operational guidance as necessary to configure the TSF to negotiate TLS 1.2 and reject renegotiation. The evaluator shall initiate a valid initial TLS 1.2 session, send a valid ClientHello on the non-renegotiable TLS channel, and observe that the TSF does not perform renegotiation of the TLS channel. **TD0899 has been applied.**
Test Steps	• Send a valid ClientHello message over the non-renegotiable TLS channel to start a session using the specified TLS version and monitor the connection to ensure the TOE terminates the session after receiving the renegotiation request. • Verify through packet capture that the TOE terminates the session after receiving the ClientHello message.
Pass/Fail with Explanation	Pass. The TOE correctly negotiates the specified TLS version and terminates the session when renegotiation is attempted. This meets the testing requirements.

6.1.4 IDENTIFICATION AND AUTHENTICATION (FIA)

6.1.4.1 FIA_AFL.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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): 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 reenabled). The evaluator shall test that once the authentication attempts limit is reached, authentication attempts with valid credentials are no longer successful.
Test Steps	• Configure the unsuccessful authentication attempts threshold and lockout period. • Start a session with the TOE and attempt to login with the wrong password and lock the user. • Verify the user is locked out for configured time with logs. • Attempt to open another connection and attempt to login with valid password before the lockout period expires. • Verify with logs the attempt failed due to locked out account.
Pass/Fail with Explanation	Pass. The TOE does not allow authentication once the unsuccessful authentication attempt limit has been reached. This meets the testing requirement.

6.1.4.2 FIA_AFL.1 TEST #2A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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): 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).
Pass/Fail with Explanation	NA. Administrator action is not selected in the ST, hence this test is not applicable.

6.1.4.3 FIA_AFL.1 TEST #2B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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): Test 2: After reaching the limit for unsuccessful authentication attempts as in Test 1 above, the evaluator shall proceed as follows. 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	• Configure the unsuccessful authentication attempts threshold and lockout period. • Start a session with the TOE and attempt to login with wrong passwords and lockout the user. • Verify with logs that the user has been locked out. • Verify that the lockout time matches the configured lockout period. • Confirm that an attempt to establish a session with correct credentials just before the end of the lockout period fails. • Verify that the user gets unlocked after the end of the lockout period. • Confirm that an attempt to establish a session with correct credentials succeeds now.
Pass/Fail with Explanation	Pass. The TOE does not allow authentication until the configured lock-out time period has expired. Once the configured time period has elapsed, the TOE allows the administrator to login. This meets the testing requirement.

6.1.4.4 FIA_PMG_EXT.1 TEST #1 (CPP_ND_V3.0E)

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

Test Steps	• Set the minimum password requirements.
	  ○ Minimum 15-character length.
	  ○ Minimum 4 upper case.
	  ○ Minimum 4 lower case.
	  ○ Minimum 3 digits.
	  ○ Minimum 3 special characters.

	• Create below users having 15-character password with combination of uppercase, lowercase, numbers and characters.
	  ○ username: good001 & password: AB1CD7+!a@bc1d]
	  ○ username: good002 & password: FG2HI8-#f$gh2i[
	  ○ username: good003 & password: KL3MN9/%k^lm3n>
	  ○ username: good004 & password: PQ4RS0&p*qr4s<~
	  ○ username: good005 & password: UV5WX1(u'vw5x>
	  ○ username: good006 & password: ZT6B2D)z=_b6cd(
	  ○ username: good007 & password: E1Jj?[Oo8]/tY4y
	  ○ username: good008 & password: Aa3b'(5cD)9~Ete
	• Show that all the users were created.

Pass/Fail with Explanation	Pass. The TOE allows configuring passwords meeting the requirements. This meets the testing requirement.

6.1.4.5 FIA_PMG_EXT.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests. 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	• Attempt to create below 5 users with passwords that do not meet the password requirements. The TOE will not allow the creation of these accounts as they did not meet the password length or complexity set by the TOE.
	  ○ Username: bad001 and Password: ef1gh7+!a@bc1d,
	  ○ Username: bad002 and Password: FG2HI8-#A$BC2D.
	  ○ Username: bad003 and Password: KLaMNb/%k^lmcn:
	  ○ Username: bad004 and Password: PQ4RS012p3qr4s
	  ○ Username: bad005 and Password: UV5WX1=(u)vw5x

Pass/Fail with Explanation	Pass. The TOE does not allow to create a user with a weak password that does not meet the password combination requirements. This meets the test requirement.

6.1.4.6 FIA_UIA_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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:

	Test 1: The evaluator shall use the guidance documentation to configure the appropriate credential supported for the login method. For all combinations of supported credentials and login methods, 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	Local (password-based) • Attempt to log into the TOE via console with bad credentials; this should fail with appropriate TOE logs. • Attempt to log into the TOE via console with good credentials; this should succeed with appropriate TOE logs. Remote (password-based) • Log into the TOE with bad credentials; this should fail. • Log into the TOE with good credentials; this should succeed. • Verify audit logs reflect both attempts.
Pass/Fail with Explanation	Pass. • The TOE denies remote (password-based) user authentication with incorrect credentials but successfully authenticates with correct credentials.

6.1.4.7 FIA_UIA_EXT.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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: 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	Remote (password-based) • Show that commands are not available prior to login with authentication logs reflecting failure. • Verify authentication logs reflect failure. • Verify that only the login banner was displayed and observe that TOE allows the establishment of a session with a remote management station prior to the identification and authentication process. • Login into the TOE. • Show that the previously disabled commands are now available. • Verify authentication logs reflect success.
Pass/Fail with Explanation	Pass. The TOE only displays the warning banner before authenticating the remote (password-based) user.

6.1.4.8 FIA_UIA_EXT.1 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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: 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	Local • Show that commands are not available prior to login. • Verify authentication logs reflect failure. • Verify that only the login banner was displayed. • Login into the TOE. • Show that the previously enabled commands are now available. • Verify authentication logs reflect success.
Pass/Fail with Explanation	Pass. The TOE only displays the warning banner before authenticating the local console user using password. This meets the testing requirement.

6.1.4.9 FIA_UIA_EXT.1 TEST #4 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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: Test 4: For distributed TOEs where not all TOE components support the authentication of Security Administrators according to FIA_UIA_EXT.1, the evaluator shall test that the components authenticate Security Administrators as described in the TSS.
Pass/Fail with Explanation	Pass. This test has been exercised with FIA_UIA_EXT.1 test 1, 2, and 3.

6.1.4.10 FIA_UAU.7 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each method of local login allowed: 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	• Connect to the TOE via console with incorrect authentication credentials and verify that the most obscured feedback is provided. • Verify the login attempt using TOE logs. • Connect to the TOE via console with correct authentication credentials and verify that the most obscured feedback is provided. • Verify the successful login attempt using TOE logs.
Pass/Fail with Explanation	Pass. The TOE gives obscured feedback after entering the correct credentials during local console login using password. The TOE gives obscured feedback after entering the incorrect credentials during local console login using password.

6.1.4.11 FIA_X509_EXT.1.1/REV TEST #1A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). 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: 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	• Configure the TOE to connect to the TLS server. • Create a complete certificate chain. • Upload the complete certificate chain required for validation to the TOE. • Initiate a connection from the TOE to the TLS server and verify that the connection is successful (i.e., the full certificate chain is properly validated). • Verify the successful connection through a packet capture.

Pass/Fail with Explanation	Pass. The TOE successfully establishes a connection with the TLS server when a complete certificate trust chain is present. This meets the testing requirements.

6.1.4.12 FIA_X509_EXT.1.1/REV TEST #1B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). 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: 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	• To break the trust chain used in Test1a, remove the ICA certificate from the TOE's trust store. • Establish a TLS connection and verify that the connection fails. • Check the TOE logs for an error indicating "CA chain was not found". • Verify the TLS handshake failure in the packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when an incomplete certificate trust chain is present. This meets the test requirements.

6.1.4.13 FIA_X509_EXT.1.1/REV TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). 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: Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.
Test Steps	Part1: • Create a server certificate that is expired. • Show the clock on the TOE. • Initiate a TLS connection to the server from TOE with an expired server certificate and verify that it fails. • Verify the failure logs on the device, showing connection is not established due to an expired certificate. • Verify the connection is unsuccessful via packet capture. Part2: • Create a server certificate that is expired. • Show the clock on the TOE. • Attempt to upload the expired server certificate on the TOE and verify that the TOE discards the certificate. • Verify that the TOE logs indicate the certificate was rejected due to expiration.
Pass/Fail with Explanation	Pass. The TOE rejects connections when an expired certificate is presented and does not allow the upload of expired certificates. This meets the test requirements.

6.1.4.14 FIA_X509_EXT.1.1/REV TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). 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: 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 shall then attempt 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	Valid Certificate: • Use the XCA tool to generate a 3-length certificate chain and CRL. • Configure the TOE for the Syslog server and CRL checking. • Load the root CA and ICA on the TOE. • Initiate a TLS connection to the server from TOE and verify that it is successful. • Verify on the CRL server that the TOE tries to fetch the CRL. • Verify the successful connection logs on the TOE. • Verify the successful connection with packet capture. Revoked End Entity Certificate: • Revoke the End Entity certificate. • Load the root CA on the TOE. • Initiate a TLS connection to the server from TOE and verify that it fails. • Verify on the CRL server that the TOE tries to fetch the CRL. • Verify the failure logs on the TOE. • Verify the unsuccessful connection with packet capture. Revoked Peer Intermediate CA Certificate: • Revoke the peer ICA certificate. • Initiate a TLS connection to the server from TOE and verify that it fails. • Verify on the CRL server that the TOE tries to fetch the CRL. • Verify the failure logs on the TOE. • Verify the unsuccessful connection with packet capture.
Pass/Fail with Explanation	Pass. The TOE supports only CRL for certificate validation and successfully performs X.509 certificate validation using CRL. It establishes connections when valid certificates are used and rejects connections with revoked certificates. This behavior meets the testing requirements.

6.1.4.15 FIA_X509_EXT.1.1/REV TEST #4A (CPP_ND_V3.0E)

Item	Data

Test Assurance Activity	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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). 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: Test 4a: [conditional] If OCSP is selected, the evaluator shall configure an authorized responder or use a man-in-the-middle tool to use a delegated OCSP signing authority to respond to the TOE's OCSP request. The resulting positive OCSP response (certStatus: good (0)) shall be signed by an otherwise valid and trusted certificate with the extendedKeyUsage extension that does not contain the OCSPSigning (OID 1.3.6.1.5.5.7.3.9). The evaluator shall verify that the TSF does not successfully complete the revocation check. Note: Per RFC 6960 Section 4.2.2.2, the OCSP signature authority is delegated when the CA who issued the certificate in question is NOT used to sign OCSP responses.
Pass/Fail with Explanation	NA. OCSP is not selected in the ST.

6.1.4.16 FIA_X509_EXT.1.1/REV TEST #4B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). 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:

	Test 4b: [conditional] If CRL is selected, the evaluator shall present an otherwise valid CRL signed by a trusted certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.
Test Steps	• Generate an ICA certificate that does NOT have the CRLsign key usage bit set. • Load this certificate chain onto the TOE. • Initiate a TLS connection to the server from TOE and verify that it fails. • Verify validation of certificate failed as CA certificate doesn't have the CRLsign key usage bit set via TOE logs. • Verify the unsuccessful TLS connection with the help of packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects connections when the intermediate CA certificate does not have CRLsign key usage bit set. This meets the testing requirements.

6.1.4.17 FIA_X509_EXT.1.1/REV TEST #5 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). 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: 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	• Start the server using the 'acumen-tls' tool with the necessary modification to the server certificate and verify that the connection fails. • Verify error logs are generated on the TOE as a certificate with modified bytes is presented. • Verify the unsuccessful connection with packet capture.
Pass/Fail with Explanation	Pass. The evaluator modified the first eight bytes of the certificate being presented by the server and ensured that the certificate fails to validate, and the TLS handshake fails. This meets the test requirements.

6.1.4.18 FIA_X509_EXT.1.1/REV TEST #6 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). 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: Test 6: The evaluator shall modify any byte in the certificate signatureValue field (see RFC 5280 Section 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	• Using the ASN editor modify the bytes in the signatureValue field of the certificate. • Try to establish a TLS connection with the modified certificate and verify that the connection fails. • Verify the error with logs on the device showing certificate signature failure. • Verify the unsuccessful connection with packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects connections when the byte in the certificate signatureValue field is modified. This meets the test requirements.

6.1.4.19 FIA_X509_EXT.1.1/REV TEST #7 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). 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:

	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	• Start the server using the 'acumen-tls' tool with the modified public key in the certificate. • Verify the error logs on the device showing certificate verification failure. • Verify the unsuccessful connection with packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects connections when any byte is the public key of the certificate is modified. This meets the test requirements.

6.1.4.20 FIA_X509_EXT.1.1/REV TEST #8A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). 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: Test 8 (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The following tests are run when a minimum certificate path length of three certificates is implemented: 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.
Pass/Fail with Explanation	N/A. This test is not applicable since the TOE does not process CA certificates presented in certificate message.

6.1.4.21 FIA_X509_EXT.1.1/REV TEST #8B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). 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: Test 8 (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The following tests are run when a minimum certificate path length of three certificates is implemented: 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.
Pass/Fail with Explanation	N/A. This test is not applicable since the TOE does not process CA certificates presented in certificate message.

6.1.4.22 FIA_X509_EXT.1.1/REV TEST #8C (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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 expected that either OCSP or CRL revocation checking is performed when a certificate is presented to the TOE (e.g. during authentication). 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: Test 8 (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The following tests are run when a minimum certificate path length of three certificates is implemented: 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	Part1: • Create an ICA certificate using elliptic curve parameters specified as a named curve, and ensure it is signed by a trusted EC root CA. • Attempt to add the created ICA certificate on the TOE. • Verify that the TOE accepts the certificate. • Verify TOE logs generated for successful import of the ICA certificate. Part2: • In the second part of the test, modify the Intermediate certificate with an explicit format version of the Elliptic Curve parameters in the public key information field. • Attempt to add the modified Intermediate certificate on the TOE. • Verify that the TOE discards the certificate. • Verify that the certificate is rejected due to an invalid public key through TOE logs.

Pass/Fail with Explanation	Pass. The TOE accepts an ICA certificate when the public key uses named curve parameters, but rejects the certificate when the public key information is modified to use explicit curve parameters. This meets the testing requirements.

6.1.4.23 FIA_X509_EXT.1.2/REV TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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). Test 1: The evaluator shall ensure that at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator shall confirm 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). 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	The option to use X.509 certificates for trusted updates is not selected in the ST. • Create an ICA with no basicConstraint. • Attempt to upload ICA with no basicConstraint to TOE's trust store. • Verify through the logs that the TOE discards the certificate.
Pass/Fail with Explanation	Pass. The TOE rejects CA certificates that do not contain the basicConstraints extension. This meets the test requirements.

6.1.4.24 FIA_X509_EXT.1.2/REV TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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). 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 shall confirm 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	The option to use X.509 certificates for trusted updates is not selected in the ST. • Create an ICA with basicConstraint flag set to FALSE. • Attempt to upload ICA with basicConstraint flag set to FALSE to TOE's trust store. • Verify through the logs that the TOE discards the certificate.
Pass/Fail with Explanation	Pass. The TOE rejects CA certificates that contains basicConstraints extension set to false. This meets the test requirements.

6.1.4.25 FIA_X509_EXT.1.1/ITT TEST #1A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not

	sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: 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	• Import the CA, ICA certificate for TOE and ICA certificate for the PEER. • Configure the TOE for IKE/IPsec connection with the PEER (Strongswan). • Configure the PEER (Strongswan) for IKE/IPsec connection with the TOE. • Attempt to establish a connection between the TOE and PEER (Strongswan) and verify that the connection is established due to a valid chain of certificates. • Verify that the connection is successful via audit logs. • Verify that the connection is successful via packet capture.
Pass/Fail with Explanation	Pass. The TOE successfully establishes a connection with the TLS server when a complete certificate trust chain is present. This meets the testing requirements.

6.1.4.26 FIA_X509_EXT.1.1/ITT TEST #1B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.:

	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	• To break the trust chain used in Test1a, delete VM's intermediate certificate from the TOE's truststore. • Attempt to establish a connection between the TOE and PEER (Strongswan) and verify that the connection is not established due to an invalid chain of certificates. • Verify that the connection fails via audit logs. • Verify that the connection failed via packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects the connection when an incomplete certificate trust chain is present. This meets the test requirements.

6.1.4.27 FIA_X509_EXT.1.1/ITT TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.
Test Steps	Part1: Pass expired end entity through IPSEC connection: • Generate a PEER certificate that has expired. • Initiate an IPsec connection and verify that the connection fails since the PEER certificate is expired. • Verify that no tunnel is established. • Verify connection failure logs. • Verify the unsuccessful connection via Packet capture.

	Part2: • Create a server certificate that is expired. • Show the clock on the TOE. • Attempt to upload the expired server certificate on the TOE and verify that the TOE discards the certificate. • Verify that the TOE logs indicate the certificate was rejected due to expiration.
Pass/Fail with Explanation	Pass. The TOE rejects connections when an expired certificate is presented and does not allow the upload of expired certificates. This meets the test requirements.

6.1.4.28 FIA_X509_EXT.1.1/ITT TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 3: The evaluator shall test that the TOE can properly handle revoked certificates-–depending on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the TOE certificate and revocation of the TOE intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator shall then attempt the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. No testing is required if no revocation method is selected. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.
Test Steps	• Verify the OCSP checking is enabled on the TOE. Successful Connection: • Import the CA, ICA certificate for TOE and ICA certificate for the PEER. • Configure the TOE for IKE/IPsec connection with the PEER (Strongswan). • Configure the PEER (Strongswan) for IKE/IPsec connection with the TOE.

	• Generate and export the index file. Verify that all the certificates are valid in the OCSP responder index file. • Enable the OCSP responder using OpenSSL. • Attempt to establish a connection between the TOE and PEER (Strongswan) and verify that the connection is established due to a valid chain of certificates. • Verify that the connection is successful via audit logs. • Verify that the connection is successful via packet capture. Revoked TOE End Entity certificate: • Revoke the TOE end entity certificate. • Generate and export the index file. Verify that the peer certificate is revoked in the OCSP responder index file. • Enable the OCSP responder using openSSL. • Attempt to establish a connection between the TOE and PEER (Strongswan) and verify that the connection is unsuccessful due to a revoked certificate. • Verify that the connection is unsuccessful via audit logs. • Verify that the connection is unsuccessful via packet capture. Revoked ICA certificate: • Unrevoked the previously revoked TOE End Entity and revoke the ICA certificate. • Generate and export the index file. Verify that the peer intermediate certificate is revoked in the OCSP responder index file. • Enable the OCSP responder using openSSL. • Attempt to establish a connection between the TOE and PEER (Strongswan) and verify that the connection is unsuccessful due to a revoked certificate. • Verify that the connection is unsuccessful via audit logs. • Verify that the connection is unsuccessful via packet capture.
Pass/Fail with Explanation	Pass. The TOE successfully connects with unrevoked certificates and rejects connections with revoked certificates. This meets the testing requirements.

6.1.4.29 FIA_X509_EXT.1.1/ITT TEST #4A (CPP_ND_V3.0E)

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

	certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 4a: [conditional] If OCSP is selected, the evaluator shall configure an authorized responder or use a man-in-the-middle tool to use a delegated OCSP signing authority to respond to the TOE's OCSP request. The resulting positive OCSP response (certStatus: good (0)) shall be signed by an otherwise valid and trusted certificate with the extendedKeyUsage extension that does not contain the OCSPSigning (OID 1.3.6.1.5.5.7.3.9). The evaluator shall verify that the TSF does not successfully complete the revocation check. Note: Per RFC 6960 Section 4.2.2.2, the OCSP signature authority is delegated when the CA who issued the certificate in question is NOT used to sign OCSP responses.
Test Steps	• The evaluator shows the EKU of the certificate for the OCSP responder. • The evaluator attempts to make a connection to the TOE. • The evaluator shows the log evidence of the connection attempt. • The evaluator shows the connection attempt with packet capture evidence.
Pass/Fail with Explanation	Pass. The TOE rejects connections when the delegated signer certificate in OCSP is invalid and does not have OCSP-signer EKU. This meets the testing requirements.

6.1.4.30 FIA_X509_EXT.1.1/ITT TEST #4B (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 4b: [conditional] If CRL is selected, the evaluator shall present an otherwise valid CRL signed by a trusted certificate that does not have the cRLsign key usage bit set and verify that validation of the CRL fails.

Pass/Fail with Explanation	NA. CRL is not selected in the ST, hence this test is not applicable.

6.1.4.31 FIA_X509_EXT.1.1/ITT TEST #5 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)
Test Steps	• Use the acumen-strongswan tool to initiate an IPsec connection after modifying a byte in the first 8 bytes of the certificate and verify that the connection fails. • Verify that no tunnel is established. • Verify via logs that an error is generated while establishing the IPsec tunnel. • Verify via packet capture that the session is not established.
Pass/Fail with Explanation	Pass. TOE rejects connections when the first 8 bytes of the certificate are modified. This meets the test requirements.

6.1.4.32 FIA_X509_EXT.1.1/ITT TEST #6 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.:

	Test 6: The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)
Test Steps	• Use the acumen-strongswan tool to initiate an IPsec connection after modifying any byte in the certificate's signatureValue field and verify that the connection fails. • Verify that no tunnel is established. • Verify via logs that an error is generated while establishing the IPsec tunnel. • Verify via packet capture that the session is not established.
Pass/Fail with Explanation	Pass. The TOE rejects connections when presented with a certificate that has any byte modified in the certificate's signatureValue field. This meets the test requirements.

6.1.4.33 FIA_X509_EXT.1.1/ITT TEST #7 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: 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	• Use the acumen-strongswan tool to initiate an IPsec connection after modifying any byte in the certificate's public key and verify that the connection fails. • Verify that no tunnel is established. • Verify via logs that an error is generated while establishing the IPsec tunnel. • Verify via packet capture that the session is not established.
Pass/Fail with Explanation	Pass. The TOE rejects connections when any byte is the public key of the certificate is modified. This meets the test requirements.

6.1.4.34 FIA_X509_EXT.1.1/ITT TEST #8A (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The following tests are run when a minimum certificate path length of three certificates is implemented: 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.
Pass/Fail with Explanation	N/A. The TOE does not accept the presented PEER ICA certificate. Refer to test case FIA_X509_EXT.1.1/REV TEST #1B for evidence.

6.1.4.35 FIA_X509_EXT.1.1/ITT TEST #8B (CPP_ND_V3.0E)

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

	implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The following tests are run when a minimum certificate path length of three certificates is implemented: 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.
Pass/Fail with Explanation	N/A. The TOE does not accept the presented PEER ICA certificate. Refer to test case FIA_X509_EXT.1.1/REV TEST #1B for evidence.

6.1.4.36 FIA_X509_EXT.1.1/ITT TEST #8C (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.: Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The following tests are run when a minimum certificate path length of three certificates is implemented:

	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	Part1: • Create an ICA certificate using elliptic curve parameters specified as a named curve, and ensure it is signed by a trusted EC root CA. • Attempt to add the created ICA certificate on the TOE. • Verify that the TOE accepts the certificate. • Verify TOE logs generated for successful import of the ICA certificate. Part2: • In the second part of the test, modify the Intermediate certificate with an explicit format version of the Elliptic Curve parameters in the public key information field. • Attempt to add the modified Intermediate certificate on the TOE. • Verify that the TOE discards the certificate. • Verify that the certificate is rejected due to an invalid public key through TOE logs.
Pass/Fail with Explanation	Pass. The TOE accepts an ICA certificate when the public key uses named curve parameters, but rejects the certificate when the public key information is modified to use explicit curve parameters. This meets the testing requirements.

6.1.4.37 FIA_X509_EXT.1.2/ITT TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for FIA_X509_EXT.1.2/ITT. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/ITT. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted.

	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 two certificates: a self-signed root CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain). Test 1: The evaluator shall ensure that one CA in the chain does not contain the basicConstraints extension. The evaluator shall confirm that the TOE rejects such a certificate at each of the following points supported: (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 an ICA with no basicConstraint. • Attempt to upload ICA with no basicConstraint to TOE's trust store. • Verify through the logs that the TOE discards the certificate.
Pass/Fail with Explanation	Pass. The TOE rejects CA certificates that do not contain the basicConstraints extension. This meets the test requirements.

6.1.4.38 FIA_X509_EXT.1.2/ITT TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests for FIA_X509_EXT.1.2/ITT. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/ITT. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the

	TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted. 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 two certificates: a self-signed root CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain). 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 shall confirm 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).
Test Steps	The option to use X.509 certificates for trusted updates is not selected in the ST. • Create an ICA with basicConstraint flag set to FALSE. • Attempt to upload ICA with basicConstraint flag set to FALSE to TOE's trust store. • Verify through the logs that the TOE discards the certificate.
Pass/Fail with Explanation	Pass. The TOE rejects CA certificates that do not contain the basicConstraints extension. This meets the test requirements.

6.1.4.39 FIA_X509_EXT.2 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following test for each trusted channel:

	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	Part1: TOE connects with the CRL server to validate certs when attempting a TLS connection – Covered in FIA_X509_EXT.1.1/Rev Test #3 Part2: • Verify that the Root and ICA certificates have the CRL Distribution Point configured. • Manipulate the environment so that the TOE is unable to verify the validity of the certificate from the CRL server. • Initiate a TLS connection to the server from TOE and verify that the connection is unsuccessful. • Verify through the logs that CRL checking failed. • Verify an unsuccessful connection with packet capture.
Pass/Fail with Explanation	Pass. The TOE rejects the connection despite loss of CRL server availability. This meets the testing requirements.

6.1.4.40 FIA_X509_EXT.3 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: 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	• From the TOE, generate a CSR. • Examine the CSR contents. Ensure the CSR contains the following fields.

	○ Public key ○ Common Name
Pass/Fail with Explanation	Pass. The TOE is able to generate a CSR with all of the requisite information. This meets the testing requirements.

6.1.4.41 FIA_X509_EXT.3 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: 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	Part 1 • Generate a CSR (Certificate Signing Request) on the TOE. • Generate a signed certificate based on the generated CSR from an external ICA. • Ensure that the full trust chain for the signed CA is not present on the TOE i.e. only the CA certificate is added. • Attempt to load the signed certificate on the TOE. • Verify the error logs generated. Part 2 • Add the intermediate certificate to the TOE certificate store to ensure that the TOE has a full certificate path. • Verify from the logs that the intermediate certificate is installed.
Pass/Fail with Explanation	Pass. The TOE only installs a CSR response signed by a CA with a full trust path and does not validate a signed CSR if the full trust chain is not present. This meets the testing requirements.

6.1.4.42 FIA_X509_EXT.1/REV TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	There is no change to the Evaluation Activities specified for this SFR in the NDcPP Supporting Document. The PP-Module modifies this SFR to make it mandatory because of the TOE's required support for IPsec.
Pass/Fail with Explanation	Pass. This test is covered by FIA_X509_EXT.1/Rev Test#1 from the X509 module in the NDcPPv3.0e TR.

6.1.4.43 FIA_X509_EXT.2 TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	There is no change to the Evaluation Activities specified for this SFR in the NDcPP Supporting Document. The PP-Module modifies this SFR to support its use for IPsec at a minimum. The evaluator shall ensure that all evaluation of this SFR is performed against its use in IPsec communications as well as any other supported usage.
Pass/Fail with Explanation	Pass. This test is covered by FIA_X509_EXT.2 Test#1 from the X509 module in the NDcPPv3.0e TR.

6.1.4.44 FIA_X509_EXT.3 TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	There is no change to the Evaluation Activities specified for this SFR in the NDcPP Supporting Document. The PP-Module modifies this SFR to make it mandatory because of the TOE's required support for IPsec.
Pass/Fail with Explanation	Pass. This test is covered by FIA_X509_EXT.3 Test#1 from the X509 module in the NDcPPv3.0e TR.

6.1.4.45 FIA_PSK_EXT.1 TEST #1 (MOD_VPNGW_V1.3) **[TD1034]**

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE). Test FIA_PSK_EXT.1:1: For each mechanism selected in FIA_PSK_EXT.1.2 the evaluator shall attempt to establish a connection and confirm that the connection requires the selected factors in the PSK to establish the connection. **TD1034 has been applied.**
Test Steps	• Configure the TOE and peer to use IKE with PSK-based authentication. • Initiate IKE negotiation from the peer to the TOE. • Verify that the connection is successfully established via logs. • Verify that the traffic is encrypted with IPsec via PCAP.
Pass/Fail with Explanation	Pass. The successfully establishes a IPSec connection and the connection uses the selected factors in the PSK. This meets the testing requirements.

6.1.4.46 FIA_PSK_EXT.1 TEST #1 (MOD_VPNGW_V1.3) **[TD1034]**

Item	Data
Test Assurance Activity	The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE). Test FIA_PSK_EXT.1:1: For each mechanism selected in FIA_PSK_EXT.1.2 the evaluator shall attempt to establish a connection and confirm that the connection requires the selected factors in the PSK to establish the connection in alignment with table 1 from RFC 8784. **TD0838 has been applied.**
Test Steps	• Configure the TOE and peer to use IKE with PSK-based authentication. • Initiate IKE negotiation from the peer to the TOE. • Verify that the connection is successfully established via logs. • Verify that the traffic is encrypted with IPsec via PCAP.
Pass/Fail with Explanation	Pass. The successfully establishes a IPSec connection and the connection uses the selected factors in the PSK in alignment with table 1 from RFC 8784. This meets the testing requirements.

6.1.4.47 FIA_PSK_EXT.2 TEST #1 (MOD_VPNGW_V1.3) **[TD1034]**

Item	Data
Test Assurance Activity	Test FIA_PSK_EXT.2:1: [conditional] If generate was selected the evaluator shall generate a pre-shared key and confirm the output matches the size selected in FIA_PSK_EXT.2.1.
Pass/Fail with Explanation	NA. The ST does not select generate pre-shared keys; hence this activity is not applicable.

6.1.5 SECURITY MANAGEMENT (FMT)

6.1.5.1 FMT_MOF.1/MANUALUPDATE TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: 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.

Test Steps	• Create a read only user to attempt to perform an update using a legitimate update image. • Verify that the user “tester” fails to update the TOE, as it is not authorized to change the settings.
Pass/Fail with Explanation	Pass. The TOE does not allow a non-security administrator to update it with a legitimate image without proper authentication. This meets the testing requirements.

6.1.5.2 FMT_MOF.1/MANUALUPDATE TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: 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
Pass/Fail with Explanation	Pass. This testing is covered by the requirements in FPT_TUD_EXT.1 Test #1.

6.1.5.3 FMT_MOF.1/SERVICES TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) without prior authentication as Security Administrator (either by authenticating as a user with no administrator privileges, if possible, or without prior authentication at all). The attempt to enable/disable this service/these services should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to enable/disable this service/these services can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator.
Test Steps	• Login to the TOE with lower privilege user.

	• Make sure the user is non-privileged user. • Attempt to enable and disable the services and verify that the user is unable to do so.
Pass/Fail with Explanation	Pass. TOE does not allow a non-security administrator to enable or disable any services. This meets the testing requirement.

6.1.5.4 FMT_MOF.1/SERVICES TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: The evaluator shall try to enable and disable at least one of the services as defined in the Application Notes for FAU_GEN.1.1 (whichever is supported by the TOE) with prior authentication as Security Administrator. The attempt to enable/disable this service/these services should be successful.
Test Steps	• Login to the TOE with the admin user. • Attempt to enable and disable the services and observe it succeed. • Verify using the TOE logs that security administrator was able to enable or disable services.
Pass/Fail with Explanation	Pass, The TOE allows modification of services from an administrative user from a local interface but not the HTTPS interface.

6.1.5.5 FMT_MTD.1/COREDATA TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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.
Pass/Fail with Explanation	Pass. All testing for all management functions have been tested under the table found in FMT_SMF.1 Test #1.

6.1.5.6 FMT_MTD.1/CRYPTOKEYS TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: 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.
Test Steps	• Start a session on the TOE with a non-administrator user. • Attempt to modify cryptographic keys i.e. delete/modify/generate/import CSR. This will fail.
Pass/Fail with Explanation	Pass. The test passes because the cryptographic key was not able to be generated due to the lack of permissions of the non-admin user.

6.1.5.7 FMT_MTD.1/CRYPTOKEYS TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: The evaluator shall try to perform at least one of the related actions with prior authentication as Security Administrator. This attempt should be successful.
Pass/Fail with Explanation	Pass. This test has been excercised in FIA_X509_EXT.1.1/Rev Test #1a.

6.1.5.8 FMT_MTD.1/CRYPTOKEYS TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	There is no change to the Evaluation Activities specified for this SFR in the NDcPP Supporting Document. The PP-Module modifies this SFR to make it mandatory because of the TOE's required support for IPsec.
Pass/Fail with Explanation	Pass. This test is covered by FMT_MTD.1/Cryptokeys Test#1 from the Auth module in the NDcPPv3.0e TR.

6.1.5.9 FMT_SMF.1/VPN TEST #1 (MOD_VPNGW_V1.3)

Item	Data

Test Assurance Activity	The evaluator tests management functions as part of performing other test EAs. No separate testing for FMT_SMF.1/VPN is required unless one of the management functions in FMT_SMF.1.1/VPN has not already been exercised under any other SFR.
Test Steps	The TSF shall be capable of performing the following management functions: [• Definition of packet filtering rules • Association of packet filtering rules to network interfaces • Ordering of packet filtering rules by priority
Pass/Fail with Explanation	Pass. All management functions identified have been tested throughout the evaluation. Thus, this requirement has been met.

6.1.5.10 FMT_SMF.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall test 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.

Test Steps	• Ability to administer the TOE remotely;
	• Ability to configure the access banner;
	• Ability to configure the remote session inactivity time before session termination;
	• Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
	• [
	  ○ Ability to start and stop services;
	  ○ Ability to modify the behaviour of the transmission of audit data to an external IT entity;
	  ○ Ability to manage the cryptographic keys;
	  ○ Ability to configure the cryptographic functionality;
	  ○ Ability to configure the lifetime for IPsec SAs;
	  ○ Ability to configure the list of supported (D)TLS ciphers;
	  ○ Ability to configure the interaction between TOE components;
	  ○ Ability to set the time which is used for time-stamps;
	  ○ Ability to configure the reference identifier for the peer;
	  ○ Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
	  ○ Ability to generate Certificate Signing Request (CSR) and process CA certificate response;
	  ○ Ability to administer the TOE locally;
	  ○ Ability to configure the local session inactivity time before session termination or locking;
	  ○ Ability to configure the authentication failure parameters for FIA_AFL.1;
	].

Pass/Fail with Explanation	Pass. All management functions identified have been tested throughout the evaluation. Thus, this requirement has been met.

6.1.5.11 FMT_SMF.1/VPN TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	The evaluator tests management functions as part of performing other test EAs. No separate testing for FMT_SMF.1/VPN is required unless one of the management functions in FMT_SMF.1.1/VPN has not already been exercised under any other SFR.
Test Steps	The TSF shall be capable of performing the following management functions: [• Definition of packet filtering rules • Association of packet filtering rules to network interfaces • Ordering of packet filtering rules by priority
Pass/Fail with Explanation	Pass. All management functions identified have been tested throughout the evaluation. Thus, this requirement has been met.

6.1.5.12 FMT_SMF.1/FFW TEST #1 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	The evaluation activities specified for FMT_SMF.1 in the Supporting Document for the Base-PP shall be applied in the same way to the newly added management functions defined in FMT_SMF.1/FFW in the FW Module.
Test Steps	The TSF shall be capable of performing the following functions: [• Ability to configure firewall rules;
Pass/Fail with Explanation	Pass. All management functions identified have been tested throughout the evaluation. Thus, this requirement has been met.

6.1.5.13 FMT_SMR.2 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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, if the TSF shall be validated against the Functional Package for Secure Shell referenced in Section 2.2 of the cPP; and

	TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.
Pass/Fail with Explanation	Pass. There are two interfaces where these can be tested (over the local console and remote web). It is covered in FIA_UIA_EXT.1.1 Test #2, FIA_UIA_EXT.1.1 Test #3, FTA_SSL_EXT.1.1 Test #1, FTA_SSL.3.1 Test #1, FTA_SSL.4.1 Test #1 and FTA_SSL.4.1 Test #2. This meets the testing requirements.

6.1.6 PROTECTION OF THE TSF (FPT)

6.1.6.1 FPT_FLS.1/SELFTEST TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	There are no test EAs for this component.
Pass/Fail with Explanation	NA. There are no test EAs for this component.

6.1.6.2 FPT_TST_EXT.1 TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	There is no change to the Evaluation Activities specified for this SFR in the NDcPP Supporting Document. The PP-Module requires a particular self-test to be performed, but this self-test is still evaluated using the same methods specified in the Supporting Document.
Pass/Fail with Explanation	Pass. This test is covered by FPT_TST_EXT.1 TEST #1 from the X509 module in the NDcPPv3.0e TR.

6.1.6.3 FPT_TST_EXT.3 TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	There are no test EAs for this component.
Pass/Fail with Explanation	NA. There are no test EAs for this component.

6.1.6.4 FPT_TUD_EXT.1 TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	There is no change to the Evaluation Activities specified for this SFR in the NDcPP Supporting Document. The PP-Module modifies this SFR to mandate that a particular selection be chosen, but this selection is part of the original definition of the SFR so no new behavior is defined by the PP-Module.
Pass/Fail with Explanation	Pass. This test is covered by the Update module in the NDcPPv3.0e TR.

6.1.6.5 FPT_APW_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	No test assurance activities have been identified.
Pass/Fail with Explanation	No test assurance activities have been identified.

6.1.6.6 FPT_ITT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: The evaluator shall ensure that communications using each protocol between each pair of authorized TOE components is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful. Further assurance activities are associated with the specific protocols.
Test Steps	• The evaluator configures the IPsec tunnel between each TOE component. • The evaluator connects each of the TOE components. • The evaluator verifies the connection attempt with packet capture evidence. • The evaluator verifies the connection attempt with log evidence.
Pass/Fail with Explanation	The TOE allows encrypted traffic to pass after being configured from guidance documentation.

6.1.6.7 FPT_ITT.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext. Further assurance activities are associated with the specific protocols.
Pass/Fail with Explanation	Pass. This test has been excercised in FPT_ITT.1 Test #1.

6.1.6.8 FPT_ITT.1 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 3: Objective: This test is to ensure that the TOE reacts appropriately to any connection outage or interruption of the route between distributed components. The evaluator shall ensure that, for each different pair of nonequivalent component types, the connection is physically interrupted for the following durations: i) a duration that exceeds the TOE's application layer timeout setting, ii) a duration that is shorter than the application layer timeout but is of sufficient length to interrupt the network link layer. The evaluator shall ensure that when physical connectivity is restored, either communications are appropriately protected, or the secure channel is terminated and the registration process (as described in the FTP_TRP.1/Join) re-initiated, with the TOE generating adequate warnings to alert the Security Administrator. In the case that the TOE is able to detect when the cable is removed from the device, another physical network device (e.g. a core switch) shall be used to interrupt the connection between the components. For a non-virtualized TOE, 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.
Test Steps	**Less than application layer timeout test (2 minutes)** **IPsec** • The evaluator authenticates the TOE. • The evaluator displays packet capture data between the TOE and IPsec client. **Greater than application layer timeout test (5 minutes)** **IPSec** • The evaluator authenticates the TOE. • The evaluator displays packet capture data between the TOE and IPsec client. **Note: The network interruption between the TOE and testing VM was caused by a network switch in between the TOE and main network switch. The network cable was unplugged from the network switch and not the TOE.**
Pass/Fail with Explanation	Pass. The TOE does not send plaintext traffic when disconnected for a short and long period of time from the IPsec client. This meets the testing requirements.

6.1.6.9 FPT_SKP_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	No test assurance activities have been identified.
Pass/Fail with Explanation	No test assurance activities have been identified.

6.1.6.10 FPT_STM_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: If the TOE supports **direct setting of the time** by the Security Administrator then the evaluator shall use the guidance documentation to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.

	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	• The evaluator displays the current time prior to the time being set. • The evaluator configures the time on the TOE. • The evaluator shows log evidence of the time being set.
Pass/Fail with Explanation	Pass, The TOE was successful in allowing the system time to be set.

6.1.6.11 FPT_STM_EXT.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: 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 shall 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.
Pass/Fail with Explanation	N/A. The TOE does not support the use of an NTP server, hence this test is not applicable.

6.1.6.12 FPT_STM_EXT.1 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	Test 3 [conditional]: If the TOE obtains time from the underlying VS, the evaluator shall record the time on the TOE, modify the time on the underlying

	VS, and verify the modified time is reflected by the TOE. If there is a delay between the setting the time on the VS and when the time is reflected on the TOE, the evaluator shall ensure this delay is consistent with the TSS and Guidance. 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	• The evaluator displays the current time on the virtual system prior to the time being set. • The evaluator configures the time on the virtual system. • The evaluator displays the current time on the TOE to reflect the virtual system time. • The evaluator shows log evidence of the time being set.
Pass/Fail with Explanation	Pass. The TOE allows the time to be configured from the virtual system.

6.1.6.13 FPT_TST_EXT.1 TEST #1 (CPP_ND_V3.0E) **[TD0836]**

Item	Data
Test Assurance Activity	It is expected that at least the following tests are performed: a) Verification of the integrity of the firmware and executable software of the TOE b) Verification of the correct operation of the cryptographic functions necessary to fulfil any of the SFRs. Although formal compliance is not mandated, the self-tests performed should aim for a level of confidence comparable to: a) [FIPS 140-2], Section 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], Section 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 verify that the self-tests described above are carried out according to the SFR and in agreement with the descriptions in the TSS. 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. **TD0836 has been applied.**
Test Steps	• Reboot the TOE and verify the device performs self-checks. • Verify after the TOE reboots, the self-checks are performed and passed.
Pass/Fail with Explanation	Pass, The TOE successfully verifies the integrity of firmware and self-tests were performed correctly.

6.1.6.14 FPT_TUD_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1: The evaluator shall perform 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 shall verify 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 shall perform 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.

	The evaluator shall perform Test 1 and Test 2 for all methods supported (manual updates, automatic checking for updates, automatic updates). For distributed TOEs the evaluator shall perform Test 1 and Test 2 for all TOE components.
Test Steps	• Verify the older version of the installed image. • Upload the image to the TOE.
Pass/Fail with Explanation	Pass. The TOE can be successfully updated with a legitimate image. This meets the testing requirements.

6.1.6.15 FPT_TUD_EXT.1 TEST #2 (A) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: 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 shall first confirm that no updates are pending and then perform 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 shall obtain or produce illegitimate updates as defined below and attempt to install them on the TOE. The evaluator shall verify that the TOE rejects all of the illegitimate updates. The evaluator shall perform this test using all of the following forms of illegitimate updates: i. A modified version (e.g. using a hex editor) of a legitimately signed update 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 the current version and most recently installed version, reflect the same version information as prior to the update attempt.

	The evaluator shall perform Test 1 and Test 2 for all methods supported (manual updates, automatic checking for updates, automatic updates). For distributed TOEs the evaluator shall perform Test 1 and Test 2 for all TOE components.
Test Steps	• Attempt to install a modified update to the TOE. • Verify the update couldn't be installed to the TOE.
Pass/Fail with Explanation	Pass, The TOE successfully rejects loading of the modified image allowing the evaluator to revert back to the previous working image.

6.1.6.16 FPT_TUD_EXT.1 TEST #2 (B) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: 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 shall first confirm that no updates are pending and then perform 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 shall obtain or produce illegitimate updates as defined below and attempt to install them on the TOE. The evaluator shall verify that the TOE rejects all of the illegitimate updates. The evaluator shall perform this test using all of the following forms of illegitimate updates: ii. An image that has not been signed 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 the current version and most recently installed version, reflect the same version information as prior to the update attempt.

	The evaluator shall perform Test 1 and Test 2 for all methods supported (manual updates, automatic checking for updates, automatic updates). For distributed TOEs the evaluator shall perform Test 1 and Test 2 for all TOE components.
Test Steps	• Attempt to install a modified update to the TOE. • Verify the update couldn't be installed to the TOE.
Pass/Fail with Explanation	Pass, The TOE successfully rejects loading of the modified image allowing the evaluator to revert back to the previous working image.

6.1.6.17 FPT_TUD_EXT.1 TEST #2 (C) (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an i confirm that no updates are pending and then perform the version verification activity to determi this test. The evaluator shall obtain or produce illegitimate updates as defined below and attempt perform this test using all of the following forms of illegitimate updates: iii. An image signed with an invalid signature (e.g. by using a different key as expected for cre The handling of version information of the most recently installed version might differ between di handles the most recently installed version information for that case as described in the guidance recently installed version, reflect the same version information as prior to the update attempt. The evaluator shall perform Test 1 and Test 2 for all methods supported (manual updates, automa For distributed TOEs the evaluator shall perform Test 1 and Test 2 for all TOE components.
Test Steps	• Attempt to install a modified update to the TOE. • Verify the update couldn't be installed to the TOE.
Pass/Fail with Explanation	Pass, The TOE successfully rejects loading of the modified image.

6.1.7 PACKET FILTERING (FPF)

6.1.7.1 FPF_RUL_EXT.1.1 TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.1:1: The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would otherwise be denied by the ruleset should be sourced and directed to a host. The evaluator shall use a packet sniffer to verify none of the generated network traffic is permitted through the TOE during initialization. Note: The remaining testing associated with application of the ruleset is addressed in the subsequent test EAs.
Test Steps	IPv4: • Configure a filter on the TOE to reject traffic from a specific IPv4 source address. • Send continuous traffic from the chosen source address and verify that it's rejected. • Verify the traffic is denied via TOE logs. • Verify the traffic is denied via packet capture. • Reboot the TOE while the ping is in progress. • Verify via TOE logs that traffic from the chosen source address was denied after reboot. • Verify with packet capture that all traffic from the chosen source address was denied during reboot. IPv6: • Configure a filter on the TOE to reject traffic from a specific IPv6 source address. • Send continuous traffic from the chosen source address and verify that it's rejected. • Verify the traffic is denied via TOE logs. • Verify the traffic is denied via packet capture. • Reboot the TOE while the ping is in progress. • Verify via TOE logs that traffic from the chosen source address was denied after reboot. • Verify with packet capture that all traffic from the chosen source address was denied during reboot.
Pass/Fail with Explanation	Pass. The IPv4 and IPv6 traffic that would otherwise be denied by the TOE was not permitted through during TOE initialization. This meets the testing requirement.

6.1.7.2 FPF_RUL_EXT.1.1 TEST #2 (MOD_VPNGW_V1.3)

Item	Data

Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.1:2: The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would be permitted by the ruleset should be sourced and directed to a host. The evaluator shall use a packet sniffer to verify none of the generated network traffic is permitted through the TOE during initialization and is only permitted once initialization is complete. Note: The remaining testing associated with application of the ruleset is addressed in the subsequent test EAs.
Test Steps	IPv4: • Configure a filter to allow traffic with a specific source address. • Send continuous traffic from the specified source address and verify it is accepted. • Reboot the TOE when ping is in progress. • Verify through the firewall log that traffic from a specific source address is allowed after the reboot. • Verify through a packet capture that all traffic is denied when the TOE is performing a reboot but once the TOE is operational all traffic from the specific source address is allowed. IPv6: • Configure a filter to allow traffic with a specific source address. • Send continuous traffic from the specified source address and verify it is accepted. • Reboot the TOE when ping is in progress. • Verify through the firewall log that traffic from a specific source address is allowed after the reboot. • Verify through a packet capture that all traffic is denied when the TOE is performing a reboot but once the TOE is operational all traffic from the specific source address is allowed.
Pass/Fail with Explanation	Pass. The TOE did not permit any network traffic during initialization and allowed it only after initialization was complete.

6.1.7.3 FPF_RUL_EXT.1.2 TEST #1 (MOD_VPNGW_V1.3)

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

Pass/Fail with Explanation	Pass. The definition of packet filtering policy, association of operations with packet filtering rules, and association of these rules to network interfaces is described collectively under FPF_RUL_EXT.1.4.

6.1.7.4 FPF_RUL_EXT.1.3 TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	There are no EAs specified for this element. Definition of packet filtering policy, association of operations with packet filtering rules, and association of these rules to network interfaces is described collectively under FPF_RUL_EXT.1.4.
Pass/Fail with Explanation	Pass. The definition of packet filtering policy, association of operations with packet filtering rules, and association of these rules to network interfaces is described collectively under FPF_RUL_EXT.1.4.

6.1.7.5 FPF_RUL_EXT.1.4 TEST #1 (MOD_VPNGW_V1.3)

Item	Data

Test Assurance Activity	The evaluator shall perform the following tests:
	
	Test FPF_RUL_EXT.1.4:1: The evaluator shall use the instructions in the operational guidance to test that packet filter rules can be created that permit, discard, and log packets for each of the following attributes:
	• IPv4
	  ○ Source address
	  ○ Destination Address
	  ○ Protocol
	• IPv6
	  ○ Source address
	  ○ Destination Address
	  ○ Next Header (Protocol)
	• TCP
	  ○ Source Port
	  ○ Destination Port
	• UDP
	  ○ Source Port
	  ○ Destination Port
	
	Note that these test activities should be performed in conjunction with those of FPF_RUL_EXT.1.6 where the effectiveness of the rules is tested; here the evaluator is just ensuring the guidance is sufficient and the TOE supports the administrator creating a ruleset based on the above attributes. The test activities for FPF_RUL_EXT.1.6 define the combinations of protocols and attributes required to be tested. If those combinations

	are configured manually, that will fulfill the objective of these test activities, but if those combinations are configured otherwise (e.g., using automation), these test activities may be necessary in order to ensure the guidance is correct and the full range of configurations can be achieved by a TOE administrator.
Test Steps	IPv4 Source address: • Configure a filter on the TOE to drop and accept traffic with specified IPv4 source addresses. • Using the 'acumen_firewall_vpn_tests' tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the packets with specified IPv4 source addresses are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. IPv4 Destination Address: • Configure a filter on the TOE to drop and accept traffic with specified IPv4 destination addresses. • Using the 'acumen_firewall_vpn_tests' tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the packets with specified IPv4 destination addresses are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. IPv4 Protocol: • Configure a filter on the TOE to drop IPv4 packets with TCP traffic and accept packets for ICMP traffic. • Using the 'acumen_firewall_vpn_tests' tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the IPV4 packets with specified transport layer protocol are either dropped or allowed. • Verify via packet capture that the IPV4 packets with specified transport layer protocol are either dropped or allowed. IPv6 Source address: • Configure a filter on the TOE to drop and accept traffic with specified IPv6 source addresses. • Using the scapy tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the packets with specified IPv6 source addresses are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. IPv6 Destination Address:

	• Configure a filter on the TOE to drop and accept traffic with specified IPv6 destination addresses. • Using the scapy tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the packets with specified IPv6 destination addresses are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. IPv6 Transport Layer Protocol: • Configure a filter on the TOE to drop IPv6 packets with UDP traffic and accept packets for TCP traffic. • Using the scapy tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the IPV6 packets with specified transport layer protocol are either dropped or allowed. • Verify via packet capture that the IPV6 packets with specified transport layer protocol are either dropped or allowed. TCP Source Port: • Configure a filter on the TOE to drop and accept traffic according to specified TCP source ports. • Using the 'acumen_firewall_vpn_tests' tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the TCP packets with specified source ports are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. TCP Destination Port: • Configure a filter on the TOE to drop and accept traffic according to specified TCP destination ports. • Using the 'acumen_firewall_vpn_tests' tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the TCP packets with specified destination ports are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. UDP Source Port: • Configure a filter on the TOE to drop and accept traffic according to specified UDP source ports. • Generate and send traffic that matches the applied filter. • Verify via TOE logs that the UDP packets with specified source ports are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. UDP Destination Port:

	• Configure a filter on the TOE to drop and accept traffic according to specified UDP destination ports. • Generate and send traffic that matches the applied filter. • Verify via TOE logs that the UDP packets with specified destination ports are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration.
Pass/Fail with Explanation	Pass. For ICMPv4, ICMPv6, TCP, UDP, IPv4 and IPv6, TOE successfully implemented full packet filter rules that permit, drop, and log packets for each of the specified attributes.

6.1.7.6 FPF_RUL_EXT.1.4 TEST #2 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.4:2: The evaluator shall repeat Test 1 above for each distinct network interface type supported by the TOE to ensure that packet filtering rules can be defined for all supported types. Note that these test activities should be performed in conjunction with those of FPF_RUL_EXT.1.6 where the effectiveness of the rules is tested; here the evaluator is just ensuring the guidance is sufficient and the TOE supports the administrator creating a ruleset based on the above attributes. The test activities for FPF_RUL_EXT.1.6 define the combinations of protocols and attributes required to be tested. If those combinations are configured manually, that will fulfill the objective of these test activities, but if those combinations are configured otherwise (e.g., using automation), these test activities may be necessary in order to ensure the guidance is correct and the full range of configurations can be achieved by a TOE administrator.
Pass/Fail with Explanation	Pass. This test requirement has been covered as part of FPF_RUL_EXT.1.4 TEST #1. (The TOE consists of only one interface type).

6.1.7.7 FPF_RUL_EXT.1.5 TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.5:1: The evaluator shall devise two equal packet filtering rules with alternate operations – permit and discard. The rules should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first rule is enforced in

	both cases by generating applicable packets and using packet capture and logs for confirmation.
Test Steps	Note: TOE does not implement a mechanism that ensures that no conflicting rules can be configured. When conflicting rules are present in the firewall filter, the TOE evaluates them sequentially, and the first matching rule dictates the action taken on the packet. IPv4: • Configure a filter to allow and drop packets to a specific destination IP with the allow rule taking precedence. • Using scapy tool, send the traffic to the configured destination address in the filter. • Verify via TOE logs that traffic to the destination IP is being allowed. • Verify via packet capture that traffic to the destination IP is being allowed. • Configure a filter to allow and drop packets to a specific destination IP with the deny rule taking precedence. • Using scapy tool, send the traffic to the configured destination address in the filter. • Verify via TOE logs that traffic to the destination IP is being denied. • Verify via packet capture that traffic to the destination IP is being denied. IPv6: • Configure a filter to allow and drop packets to a specific destination IP with the allow rule taking precedence. • Using scapy tool, send the traffic to the configured destination address in the filter. • Verify via TOE logs that traffic to the destination IP is being allowed. • Verify via packet capture that traffic to the destination IP is being allowed. • Configure a filter to allow and drop packets to a specific destination IP with the deny rule taking precedence. • Using scapy tool, send the traffic to the configured destination address in the filter. • Verify via TOE logs that traffic to the destination IP is being denied. • Verify via packet capture that traffic to the destination IP is being denied.
Pass/Fail with Explanation	Pass. The TOE enforces the first rule for both IPv4 and IPv6 when two equal stateful traffic filtering rules with conflicting operations (permit and drop) are configured and deployed in two distinct orders. This meets the testing requirement.

6.1.7.8 FPF_RUL_EXT.1.5 TEST #2 (MOD_VPNGW_V1.3)

Item	Data

Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.5:2: The evaluator shall repeat the procedure above, except that the two rules should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator shall test both orders to ensure that the first is enforced regardless of the specificity of the rule.
Test Steps	IPv4: • Configure the firewall rule so that the first rule allows packets to a specific destination address and the second rule denies packets to its network segment with the allow rule taking precedence. • Using the scapy tool, send the traffic to the configured destination address in the filter. • Verify via TOE logs that traffic to the destination address is being allowed. • Verify via packet capture that traffic to the destination address is being allowed. • Configure the firewall rule so that the first rule denies packets to a network segment and the second rule allows packets to a specific destination address of the network segment with the deny rule taking precedence. • Using the scapy tool, send the traffic to a destination address from the configured network segment in the filter. • Verify via TOE logs that traffic to the destination address is being denied. • Verify via packet capture that traffic to the destination address is being denied. IPv6: • Configure the firewall rule so that the first rule allows packets to a specific destination address and the second rule denies packets to its network segment with the allow rule taking precedence. • Using the scapy tool, send the traffic to the configured destination address in the filter. • Verify via TOE logs that traffic to the destination address is being allowed. • Verify via packet capture that traffic to the destination address is being allowed. • Configure the firewall rule so that the first rule denies packets to a network segment and the second rule allows packets to a specific destination address of the network segment with the deny rule taking precedence. • Using the scapy tool, send the traffic to a destination address from the configured network segment in the filter. • Verify via TOE logs that traffic to the destination address is being denied. • Verify via packet capture that traffic to the destination address is being denied.

Pass/Fail with Explanation	Pass. Regardless of the rule's specificity, the TOE enforces the first rule for both IPv4 and IPv6 when two rules are devised where one is a subset of the other (e.g., a specific address vs. a network segment).

6.1.7.9 FPF_RUL_EXT.1.6 TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.6:1: The evaluator shall configure the TOE to permit and log each supported IPv4 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each supported IPv4 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that the supported protocols are permitted (i.e., by capturing the packets after passing through the TOE) and logged. Any protocols not supported by the TOE must be denied. IP Transport Layer Protocols:  IP Transport Layer Protocols.xlsx
Test Steps	<u>Specific source and Specific destination:</u> • Create permit rule for specific source address and specific destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was permitted by the firewall. • Verify through packet capture that the supported protocols are permitted. <u>Specific source, wildcard destination:</u> • Create permit rules for specific source address and wildcard destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was permitted by the firewall. • Verify through packet capture that the supported protocols are permitted. <u>Wildcard source, specific destination:</u> • Create permit rules for wildcard source address and specific destination address.

	• Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was permitted by the firewall. • Verify through packet capture that the supported protocols are permitted. Wildcard source, wildcard destination: • Create permit rules for wildcard source address and wildcard destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was permitted by the firewall. • Verify through packet capture that the supported protocols are permitted.
Pass/Fail with Explanation	Pass. The TOE permits and logs packets with combinations of specific IPv4 source address and specific IPv4 destination address, specific source IPv4 address and wildcard IPv4 destination address, wildcard IPv4 source address and specific IPv4 destination address, and wildcard IPv4 source address and wildcard IPv4 destination address as configured on TOE for each defined IPV4 Transport Layer Protocol. This meets testing requirements.

6.1.7.10 FPF_RUL_EXT.1.6 TEST #2 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.6:2: The evaluator shall configure the TOE to permit all traffic except to discard and log each supported IPv4 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv4 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that the supported protocols are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Any protocols not supported by the TOE must also be denied but are not required to be logged. IP Transport Layer Protocols:  IP Transport Layer Protocols.xlsx
Test Steps	Specific source and Specific destination: • Create a deny rule for specific source address and specific destination address.

	• Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was denied by the firewall. • Verify through packet capture that packets were dropped due to the configured rule. Specific source, wildcard destination: • Create a deny rule for the specific source address and wildcard destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was denied by the firewall. • Verify through packet capture that packets were dropped due to the configured rule. Wildcard source, specific destination: • Create a deny rule for the wildcard source address and specific destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was denied by the firewall. • Verify through packet capture that packets were dropped due to the configured rule. Wildcard source, wildcard destination: • Create a deny rule for wildcard source address and wildcard destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was denied by the firewall. • Verify through packet capture that packets were dropped due to the configured rule.
Pass/Fail with Explanation	Pass. The TOE denies and logs packets with combinations of specific IPv4 source address and specific IPv4 destination address, specific IPv4 source address and wildcard IPv4 destination address, wildcard IPv4 source address and specific IPv4 destination address, and wildcard IPv4 source address and wildcard IPv4 destination address as configured on TOE for each defined IPV4 Transport Layer Protocol. This meets testing requirements.

6.1.7.11 FPF_RUL_EXT.1.6 TEST #3 (MOD_VPNGW_V1.3)

Item	Data

Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.6:3: The evaluator shall configure the TOE to permit and log each supported IPv4 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. Additionally, the evaluator shall configure the TOE to discard and log each supported IPv4 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with different (than those permitted above) combinations of a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each supported IPv4 Transport Layer Protocol and outside the scope of all source and destination addresses configured above in order to ensure that the supported protocols are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Any protocols not supported by the TOE must be denied. IP Transport Layer Protocols:  IP Transport Layer Protocols.xlsx
Test Steps	<u>Specific source and specific destination:</u> Part 1: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations: • Create a filter to permit traffic that contains a specific source address SrcA and specific destination address DstA but deny traffic that contains a specific source address SrcB and specific destination address DstA. All other traffic is discarded. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to match the filters applied to the TOE's interface. • Verify through logs that the correct traffic was permitted by the firewall and remaining traffic is discarded. • Verify through packet capture that traffic with a specific source address (SrcA) and a specific destination address (DstA) was permitted. • Verify through packet capture that traffic with a specific source address (SrcB) and a specific destination address (DstA) was denied. Part 2: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations outside the filter's scope: • Using the 'acumen_firewall_vpn_tests' tool, generate traffic with source and destination addresses outside the scope of the source and destination addresses configured in the filter.

	• Verify through logs that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. • Verify through packet capture that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. <u>Specific source, wildcard destination:</u> Part 1: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations: • Create a filter to permit traffic that contains a specific source address SrcA and wildcard destination address WC_Dst but denies traffic that contains a specific source address SrcB and wildcard destination address WC_Dst. All other traffic is discarded. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to match the filters applied to the TOE's interface. • Verify through logs that the correct traffic was permitted by the firewall and remaining traffic is discarded. • Verify through packet capture that traffic with a specific source address (SrcA) and wildcard destination address WC_Dst was permitted. • Verify through packet capture that traffic with a specific source address (SrcB) and wildcard destination address WC_Dst was denied. Part 2: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations outside the filter's scope: • Using the 'acumen_firewall_vpn_tests' tool, generate traffic with source and destination addresses outside the scope of the source and destination addresses configured in the filter. • Verify through logs that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. • Verify through packet capture that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. <u>Wildcard source, specific destination:</u> Part 1: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations: • Create a filter to permit traffic that contains a wildcard source address WC_Src and specific destination address DstA but deny traffic that contains a wildcard source address WC_Src and specific destination address DstB. All other traffic is discarded. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to match the filters applied to the TOE's interface. • Verify through logs that the correct traffic was permitted by the firewall and remaining traffic is discarded. • Verify through packet capture that traffic with a wildcard source address WC_Src and a specific destination address (DstA) was permitted.

	• Verify through packet capture that traffic with a wildcard source address WC_Src and a specific destination address (DstB) was denied. Part 2: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations outside the filter's scope: • Using the 'acumen_firewall_vpn_tests' tool, generate traffic with source and destination addresses outside the scope of the source and destination addresses configured in the filter. • Verify through logs that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. • Verify through packet capture that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. <u>Wildcard source, wildcard destination:</u> Part 1: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations: • Create a filter to permit traffic that contains a wildcard source address WC_Src and wildcard destination address WC_Dst but deny traffic that contains a wildcard source address and wildcard destination address. All other traffic is discarded. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to match the filters applied to the TOE's interface. • Verify through logs that the correct traffic was permitted by the firewall and remaining traffic is discarded. • Verify through packet capture that traffic with a wildcard source address WC_Src and wildcard destination address WC_Dst was permitted. • Verify through packet capture that traffic with a wildcard source address WC_Src and wildcard destination address WC_Dst was denied. Part 2: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations outside the filter's scope: • Using the 'acumen_firewall_vpn_tests' tool, generate traffic with source and destination addresses outside the scope of the source and destination addresses configured in the filter. • Verify through logs that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. • Verify through packet capture that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol.
Pass/Fail with Explanation	Pass. The test case showed that a rule can be configured for each of the traffic flow. Each traffic flow was logged and not permitted. This meets the testing requirement.

6.1.7.12 FPF_RUL_EXT.1.6 TEST #4 (MOD_VPNGW_V1.3)

Item	Data

Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.6:4: The evaluator shall configure the TOE to permit and log each supported IPv6 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that the supported protocols are permitted (i.e., by capturing the packets after passing through the TOE) and logged. Any protocols not supported by the TOE must be denied. IP Transport Layer Protocols:  IP Transport Layer Protocols.xlsx
Test Steps	<u>Specific source and Specific destination:</u> • Create permit rule for specific source address and specific destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was permitted by the firewall. • Verify through packet capture that the supported protocols are permitted. <u>Specific source, wildcard destination:</u> • Create permit rules for specific source address and wildcard destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was permitted by the firewall. • Verify through packet capture that the supported protocols are permitted. <u>Wildcard source, specific destination:</u> • Create permit rules for wildcard source address and specific destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was permitted by the firewall. • Verify through packet capture that the supported protocols are permitted. <u>Wildcard source, wildcard destination:</u> • Create permit rules for wildcard source address and wildcard destination address.

	• Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was permitted by the firewall. • Verify through packet capture that the supported protocols are permitted.
Pass/Fail with Explanation	Pass. The TOE permits and logs packets with combinations of specific IPv6 source address and specific IPv6 destination address, specific IPv6 source address and wildcard IPv6 destination address, wildcard IPv6 source address and specific IPv6 destination address, and wildcard IPv6 source address and wildcard IPv6 destination address as configured on TOE for each defined IPV6 Transport Layer Protocol. This meets testing requirements.

6.1.7.13 FPF_RUL_EXT.1.6 TEST #5 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.6:5: The evaluator shall configure the TOE to permit all traffic except to discard and log each supported IPv6 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and within the configured source and destination addresses in order to ensure that the supported protocols are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Any protocols not supported by the TOE must also be denied but are not required to be logged. IP Transport Layer Protocols:  IP Transport Layer Protocols.xlsx
Test Steps	<u>Specific source and specific destination:</u> • Create a deny rule for specific source address and specific destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was denied by the firewall. • Verify through packet capture that packets were dropped due to the configured rule. <u>Specific source, wildcard destination:</u>

	• Create a deny rule for the specific source address and wildcard destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was denied by the firewall. • Verify through packet capture that packets were dropped due to the configured rule. <u>Wildcard source, specific destination:</u> • Create a deny rule for the wildcard source address and specific destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was denied by the firewall. • Verify through packet capture that packets were dropped due to the configured rule. <u>Wildcard source, wildcard destination:</u> • Create a deny rule for the wildcard source address and wildcard destination address. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to hit the access rule. • Verify through logs that the correct traffic was denied by the firewall. • Verify through packet capture that packets were dropped due to the configured rule.
Pass/Fail with Explanation	Pass. The TOE denies and logs packets with combinations of specific IPv6 source address and specific IPv6 destination address, specific IPv6 source address and wildcard IPv6 destination address, wildcard IPv6 source address and specific IPv6 destination address, and wildcard IPv6 source address and wildcard IPv6 destination address as configured on TOE for each defined IPV6 Transport Layer Protocol. This meets testing requirements.

6.1.7.14 FPF_RUL_EXT.1.6 TEST #6 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.6:6: The evaluator shall configure the TOE to permit and log each supported IPv6 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. Additionally, the evaluator shall configure the TOE to discard and

	log each supported IPv6 Transport Layer Protocol (see RFC Values for IPv4 and IPv6 table for full possible list) in conjunction with different (than those permitted above) combinations of a specific source address and specific destination address, specific source address and wildcard destination address, wildcard source address and specific destination address, and wildcard source address and wildcard destination address. The evaluator shall generate packets matching each defined IPv6 Transport Layer Protocol and outside the scope of all source and destination addresses configured above in order to ensure that the supported protocols are dropped (i.e., by capturing no applicable packets passing through the TOE) and logged. Any protocols not supported by the TOE must be denied. IP Transport Layer Protocols:  IP Transport Layer Protocols.xlsx
Test Steps	<u>Specific source and specific destination:</u> Part 1: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations: • Create a filter to permit traffic that contains a specific source address SrcA and specific destination address DstA but deny traffic that contains a specific source address SrcB and specific destination address DstA. All other traffic is discarded. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to match the filters applied to the TOE's interface. • Verify through logs that the correct traffic was permitted by the firewall and remaining traffic is discarded. • Verify through packet capture that traffic with a specific source address (SrcA) and a specific destination address (DstA) was permitted. • Verify through packet capture that traffic with a specific source address (SrcB) and a specific destination address (DstA) was denied. Part 2: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations outside the filter's scope: • Using the 'acumen_firewall_vpn_tests' tool, generate traffic with source and destination addresses outside the scope of the source and destination addresses configured in the filter. • Verify through logs that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. • Verify through packet capture that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. <u>Specific source, wildcard destination:</u> Part 1: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations:

	• Create a filter to permit traffic that contains a specific source address SrcA and wildcard destination address WC_Dst but denies traffic that contains a specific source address SrcB and wildcard destination address WC_Dst. All other traffic is discarded. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to match the filters applied to the TOE's interface. • Verify through logs that the correct traffic was permitted by the firewall and remaining traffic is discarded. • Verify through packet capture that traffic with a specific source address (SrcA) and wildcard destination address WC_Dst was permitted. • Verify through packet capture that traffic with a specific source address (SrcB) and wildcard destination address WC_Dst was denied. Part 2: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations outside the filter's scope: • Using the 'acumen_firewall_vpn_tests' tool, generate traffic with source and destination addresses outside the scope of the source and destination addresses configured in the filter. • Verify through logs that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. • Verify through packet capture that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. <u>Wildcard source, specific destination:</u> Part 1: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations: • Create a filter to permit traffic that contains a wildcard source address WC_Src and specific destination address DstA but deny traffic that contains a wildcard source address WC_Src and specific destination address DstB. All other traffic is discarded. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to match the filters applied to the TOE's interface. • Verify through logs that the correct traffic was permitted by the firewall and remaining traffic is discarded. • Verify through packet capture that traffic with a wildcard source address WC_Src and a specific destination address (DstA) was permitted. • Verify through packet capture that traffic with a wildcard source address WC_Src and a specific destination address (DstB) was denied. Part 2: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations outside the filter's scope: • Using the 'acumen_firewall_vpn_tests' tool, generate traffic with source and destination addresses outside the scope of the source and destination addresses configured in the filter. • Verify through logs that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. • Verify through packet capture that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol.

	Wildcard source, wildcard destination: Part 1: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations: • Create a filter to permit traffic that contains a wildcard source address WC_Src and wildcard destination address WC_Dst but deny traffic that contains a wildcard source address and wildcard destination address. All other traffic is discarded. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to match the filters applied to the TOE's interface. • Verify through logs that the correct traffic was permitted by the firewall and remaining traffic is discarded. • Verify through packet capture that traffic with a wildcard source address WC_Src and wildcard destination address WC_Dst was permitted. • Verify through packet capture that traffic with a wildcard source address WC_Src and wildcard destination address WC_Dst was denied. Part 2: Each supported IPv4 protocol with specific/wildcard source and destination IP combinations outside the filter's scope: • Using the 'acumen_firewall_vpn_tests' tool, generate traffic with source and destination addresses outside the scope of the source and destination addresses configured in the filter. • Verify through logs that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol. • Verify through packet capture that the traffic with IP addresses outside the scope of the configured filter is discarded for each protocol.
Pass/Fail with Explanation	Pass. The TOE discards each defined IPv6 Transport Layer Protocol not matching the configured combinations of source and destination addresses. This meets testing requirements.

6.1.7.15 FPF_RUL_EXT.1.6 TEST #7 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.6:7: The evaluator shall configure the TOE to permit and log protocol 6 (TCP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination TCP ports in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged.

Test Steps	• Create a filter to configure the TOE to permit and log protocol 6 (TCP) using a selected source port, a selected destination port and a selected source and destination port combination. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to match the filter applied on the TOE. • Verify through the firewall log that the matching traffic was permitted by the firewall. • Verify through packet capture that the traffic was permitted.
Pass/Fail with Explanation	Pass. When configured with the appropriate allow rules, the TOE permits and logs TCP traffic with a specific source port, destination port, and a combination of both the source and destination port. This meets the testing requirement.

6.1.7.16 FPF_RUL_EXT.1.6 TEST #8 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.6:8: The evaluator shall configure the TOE to discard and log protocol 6 (TCP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination TCP ports in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged.
Test Steps	• Create a filter to configure the TOE to deny and log protocol 6 (TCP) using a selected source port, a selected destination port and a selected source and destination port combination. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to match the filter applied on the TOE. • Verify through the firewall log that the matching traffic was permitted was denied by the firewall. • Verify through packet capture that the traffic was denied.
Pass/Fail with Explanation	Pass. When configured with the appropriate deny rules, the TOE discards and logs TCP traffic with a specific source port, destination port, and a combination of both the source and destination port. This meets the testing requirement.

6.1.7.17 FPF_RUL_EXT.1.6 TEST #9 (MOD_VPNGW_V1.3)

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

	Test FPF_RUL_EXT.1.6:9: The evaluator shall configure the TOE to permit and log protocol 17 (UDP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination UDP ports in order to ensure that they are permitted (i.e., by capturing the packets after passing through the TOE) and logged. Here the evaluator shall ensure that the UDP port 500 (IKE) is included in the set of tests.
Test Steps	• Create a filter to configure the TOE to permit and log protocol 17 (UDP) using a selected source port, a selected destination port and a selected source and destination port combination. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to match the filter applied on the TOE. • Verify through the firewall log that the matching traffic was permitted by the firewall. • Verify through packet capture that the traffic was permitted.
Pass/Fail with Explanation	Pass. When configured with the appropriate allow rules, the TOE permits and logs UDP traffic with a specific source port, destination port, and a combination of both the source and destination port. This meets the testing requirement.

6.1.7.18 FPF_RUL_EXT.1.6 TEST #10 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test FPF_RUL_EXT.1.6:10: The evaluator shall configure the TOE to discard and log protocol 17 (UDP) using a selected source port, a selected destination port, and a selected source and destination port combination. The evaluator shall generate packets matching the configured source and destination UDP ports in order to ensure that they are denied (i.e., by capturing no applicable packets passing through the TOE) and logged. Again, the evaluator shall ensure that UDP port 500 is included in the set of tests.
Test Steps	• Create a filter to configure the TOE to deny and log protocol 17 (UDP) using a selected source port, a selected destination port and a selected source and destination port combination. • Using the 'acumen_firewall_vpn_tests' tool, generate traffic to match the filter applied on the TOE. • Verify through the firewall log that the matching traffic was permitted was denied by the firewall. • Verify through packet capture that the traffic was denied.

Pass/Fail with Explanation	Pass. When configured with the appropriate deny rules, the TOE discards and logs UDP traffic with a specific source port, destination port, and a combination of both the source and destination port. This meets the testing requirement.

6.1.8 TOE ACCESS (FTA)

6.1.8.1 FTA_SSL_EXT.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following test: Test 1: The evaluator shall follow the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator shall establish a local interactive session with the TOE. The evaluator shall then observe that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator shall then ensure that reauthentication is needed when trying to unlock the session.
Test Steps	• The evaluator configures a local time out period of 60 seconds. • The evaluator verifies the logs on the TOE for the updated session timeout. • The evaluator attempts to connect to the TOE from the local connection and allows the session to timeout. • The evaluator verifies using TOE logs that the session was terminated. • The evaluator configures a local time out period of 90 seconds. • The evaluator verifies the logs on the TOE for the updated session timeout. • The evaluator connects to the TOE from the local connection and allows the session to timeout. • The evaluator verifies using TOE logs that the session was terminated.
Pass/Fail with Explanation	Pass, The TOE successfully times out the evaluator after the configured periods of time.

6.1.8.2 FTA_SSL.3 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	For each method of remote administration, the evaluator shall perform the following test:

	Test 1: The evaluator shall follow the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator shall establish a remote interactive session with the TOE. The evaluator shall then observe that the session is terminated after the configured time period.
Test Steps	**1 Minute** • The evaluator configures the TOE with an inactivity timeout period of 1 minute. • The evaluator verifies that a log was created for configuring the timeout period. • The evaluator logs into the TOE via remote connection and allow the session to time out. • The evaluator verifies the logs for session timeout. **90 seconds** • The evaluator configures the TOE with an inactivity timeout period of 90 seconds. • The evaluator verifies that a log was created for configuring the timeout period. • The evaluator logs into the TOE via remote connection and allow the session to time out. • The evaluator verifies the logs for session timeout.
Pass/Fail with Explanation	Pass, The TOE successfully times out the remote session after being configured to 60 seconds and 90 seconds.

6.1.8.3 FTA_SSL.4 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 1 [conditional]: If the TOE supports local administration, the evaluator shall initiate an interactive local session with the TOE. The evaluator shall then follow the guidance documentation to exit or log off the session and observes that the session has been terminated.
Test Steps	• The evaluator attempts a connection to the local console. • The evaluator displays logs of the connection attempt. • The evaluator terminates the session. • The evaluator displays logs of the connection termination.
Pass/Fail with Explanation	Pass, The evaluator was able to successfully login to the local console of the TOE and logout. The logs reflected these actions accurately.

6.1.8.4 FTA_SSL.4 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: For each method of remote administration, the evaluator shall initiate an interactive remote session with the TOE. The evaluator shall then follow the guidance documentation to exit or log off the session and observes that the session has been terminated.
Test Steps	• The evaluator makes a connection attempt to the TOE remotely using the web interface. • The evaluator terminates the connection to the TOE by following the guidance documentation to exit the session. • The evaluator verifies the logs reflect that a session has been created and terminated.
Pass/Fail with Explanation	Pass, The TOE allows a successful login and can terminate the session.

6.1.8.5 FTA_TAB.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall also perform the following test: Test 1: The evaluator shall follow 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	• The evaluator configures the access banner for the TOE. • The evaluator verifies that the audit records reflect the configuration step. • The evaluator logs into the TOE via TLS. • The evaluator logs into the TOE via console.
Pass/Fail with Explanation	Pass, The TOE successfully displays the login banner for each method of access (Local and Remote) after the user authenticates.

6.1.9 TRUSTED PATH/CHANNELS (FTP)

6.1.9.1 FTP_ITC.1/VPN TEST #1 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	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: 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. 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.
Pass/Fail with Explanation	Pass. This test is covered across FCS_IPSEC_EXT.1 testing from the IPsec module in the NDcPPv3.0e TR.

6.1.9.2 FTP_ITC.1/VPN TEST #2 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	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: 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. 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.
Pass/Fail with Explanation	Pass. This test is covered across FCS_IPSEC_EXT.1 testing from the IPsec module in the NDcPPv3.0e TR.

6.1.9.3 FTP_ITC.1/VPN TEST #3 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	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: Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext. Further assurance activities are associated with the specific protocols. For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of 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.
Pass/Fail with Explanation	Pass. This test is covered across FCS_IPSEC_EXT.1 testing from the IPsec module in the NDcPPv3.0e TR.

6.1.9.4 FTP_ITC.1/VPN TEST #4 (MOD_VPNGW_V1.3)

Item	Data
Test Assurance Activity	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: 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, interrupt the connection of that IT entity for the following durations: i) a duration that exceeds the TOE's application layer timeout setting, ii) 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 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 TOE external interruption (such as a cable being physically removed or a virtual connection being disabled), another network device shall be used to interrupt the connection between the TOE and the distinct IT entity. The interruption shall be external to the TOE (i.e., by manipulating the test environment and not by TOE configuration change).

	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	**Less than application layer timeout test (2 minutes)** **IPsec** • The evaluator authenticates the TOE. • The evaluator displays packet capture data between the TOE and IPsec client. **Greater than application layer timeout test (5 minutes)** **IPSec** • The evaluator authenticates the TOE. • The evaluator displays packet capture data between the TOE and IPsec client. **Note: The network interruption between the TOE and testing VM was caused by a network switch in between the TOE and main network switch. The network cable was unplugged from the network switch and not the TOE.**
Pass/Fail with Explanation	Pass. The TOE does not send plaintext traffic when disconnected for a short and long period of time from the IPsec client. This meets the testing requirements.

6.1.9.5 FTP_ITC.1 TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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: 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. 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.
Pass/Fail with Explanation	Pass, Test has been exercised in FAU_STG_EXT.1 Test 1 using the TLS protocol to protect audit data to an audit server where the TOE is initiating the connection.

6.1.9.6 FTP_ITC.1 TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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: 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. 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.
Pass/Fail with Explanation	Pass, Test has been exercised in FAU_STG_EXT.1 Test 1 using the TLS protocol to protect audit data to an audit server where the TOE is initiating the connection.

6.1.9.7 FTP_ITC.1 TEST #3 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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: Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext. Further assurance activities are associated with the specific protocols. For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of 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.
Pass/Fail with Explanation	Pass. The test is performed in FAU_STG_EXT.1 Test#1 and FCS_TLSC_EXT.1.1 Test#1 where a successful TLS connection is established with the audit server and the channel data is encrypted.

6.1.9.8 FTP_ITC.1 TEST #4 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	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: 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, interrupt the connection of that IT entity for the following durations: iii) a duration that exceeds the TOE's application layer timeout setting, iv) 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 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 TOE external interruption (such as a cable being physically removed or a virtual connection being disabled), another network device shall be used to interrupt the connection between the TOE and the distinct IT entity. The interruption shall be external to the TOE (i.e., by manipulating the test environment and not by TOE configuration change). 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	**Less than application layer timeout test (2 minutes)** **Syslog** • The evaluator authenticates to the TOE. • The evaluator displays packet capture data between the TOE and syslog server. **Greater than application layer timeout test (5 minutes)** **Syslog** • The evaluator authenticates to the TOE. • The evaluator displays packet capture data between the TOE and syslog server. **Note: The network interruption between the TOE and testing VM was caused by a network switch in between the TOE and main network switch. The network cable was unplugged from the network switch and not the TOE.**
Pass/Fail with Explanation	Pass, The TOE successfully starts encrypted data again after connectivity gets interrupted.

6.1.9.9 FTP_TRP.1/ADMIN TEST #1 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: 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. Further assurance activities are associated with the specific protocols. For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of trusted paths to TOE components in the Security Target.
Test Steps	• The evaluator attempts to connect to the TOE via TLS. • The evaluator verifies that Wireshark shows a successful connection and it's not in plain text. • The evaluator verifies via logs that the TOE shows a successful connection.
Pass/Fail with Explanation	Pass, The TOE allows for an administrator to create a TLS connection.

6.1.9.10 FTP_TRP.1/ADMIN TEST #2 (CPP_ND_V3.0E)

Item	Data
Test Assurance Activity	The evaluator shall perform the following tests: Test 2: The evaluator shall ensure, for each communication channel, the channel data is not sent in plaintext. Further assurance activities are associated with the specific protocols. For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of trusted paths to TOE components in the Security Target.
Pass/Fail with Explanation	Pass, This test has been exercised in FTP_TRP.1/Admin Test #1.

6.1.10 FIREWALL (FFW)

6.1.10.1 FFW_RUL_EXT.1 TEST #1 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would otherwise be denied by the ruleset should be sourced and be directed at a host. The evaluator shall verify using a packet sniffer that none of the generated network traffic is permitted through the firewall during initialization.
Test Steps	IPv4: • Configure a filter on the TOE to reject traffic from a specific IPv4 source address. • Send continuous traffic from the chosen source address and verify that it's rejected. • Verify the traffic is denied via TOE logs. • Verify the traffic is denied via packet capture. • Reboot the TOE while the ping is in progress. • Verify via TOE logs that traffic from the chosen source address was denied after reboot. • Verify with packet capture that all traffic from the chosen source address was denied during reboot. IPv6: • Configure a filter on the TOE to reject traffic from a specific IPv6 source address. • Send continuous traffic from the chosen source address and verify that it's rejected. • Verify the traffic is denied via TOE logs. • Verify the traffic is denied via packet capture. • Reboot the TOE while the ping is in progress. • Verify via TOE logs that traffic from the chosen source address was denied after reboot. • Verify with packet capture that all traffic from the chosen source address was denied during reboot.
Pass/Fail with Explanation	Pass. The IPv4 and IPv6 traffic that would otherwise be denied by the TOE was not permitted through during TOE initialization. This meets the testing requirement.

6.1.10.2 FFW_RUL_EXT.1 TEST #2 (MOD_CPP_FW_V1.4E)

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

	Note: The remaining testing associated with application of the ruleset is addressed in the subsequent test evaluation activities.
Test Steps	IPv4: • Configure a filter to allow traffic with a specific source address. • Send continuous traffic from the specified source address and verify it is accepted. • Reboot the TOE when ping is in progress. • Verify through the firewall log that traffic from a specific source address is allowed after the reboot. • Verify through a packet capture that all traffic is denied when the TOE is performing a reboot but once the TOE is operational all traffic from the specific source address is allowed. IPv6: • Configure a filter to allow traffic with a specific source address. • Send continuous traffic from the specified source address and verify it is accepted. • Reboot the TOE when ping is in progress. • Verify through the firewall log that traffic from a specific source address is allowed after the reboot. • Verify through a packet capture that all traffic is denied when the TOE is performing a reboot but once the TOE is operational all traffic from the specific source address is allowed.
Pass/Fail with Explanation	Pass**.** The TOE does not permit any of the IPv4 and IPv6 generated network traffic through the firewall during the initialization, and it is only permitted once initialization is complete. This meets the testing requirement.

6.1.10.3 FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4 TEST #1 **[TD0924] (MOD_CPP_FW_V1.4E)**

Item	**Data**
Test Assurance Activity	Test 1: The evaluator shall use the instructions in the guidance documentation to test that state full packet filter firewall rules can be created that permit, drop, and log packets for each of the following attributes: • ICMPv4 ○ Type ○ Code • ICMPv6 ○ Type ○ Code • IPv4 ○ Source address ○ Destination Address

	○ Transport Layer Protocol • IPv6 ○ Source address ○ Destination Address ○ Transport Layer Protocol and where defined by the ST author, ○ Extension Header Type, Extension Header Fields • TCP ○ Source Port ○ Destination Port • UDP ○ Source Port ○ Destination Port Note that these test activities should be performed in conjunction with those of FFW_RUL_EXT.1.9 where the effectiveness of the rules is tested. The test activities for FFW_RUL_EXT.1.9 define the protocol/attribute combinations required to be tested. If those combinations are configured manually, that will fulfil the objective of these test activities, but if those combinations are configured otherwise (e.g., using automation), these test activities may be necessary in order to ensure the guidance is correct and the full range of configurations can be achieved by a TOE administrator. **TD0924 has been applied.**
Test Steps	ICMPv4 Type: • Configure a filter on the TOE to accept ICMPv4 type 8 code 0 packets but drop ICMPv4 type 3 code 0 packets. • Using the 'acumen_firewall_vpn_tests' tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the ICMPv4 packets are dropped or allowed according to the rules applied based on type. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. ICMPv4 Code: • Configure a filter on the TOE to accept ICMPv4 type 8 code 0 packets but drop ICMPv4 type 8 code 1 packets. • Using the 'acumen_firewall_vpn_tests' tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the ICMPv4 packets are dropped or allowed according to the rules applied based on type. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. ICMPv6 Type: • Configure a filter on the TOE to accept ICMPv6 type 128 code 0 packets but drop ICMPv6 type 139 code 0 packets.

	• Using the scapy tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the ICMPv6 packets are dropped or allowed according to the rules applied based on type. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. ICMPv6 Code: • Configure a filter on the TOE to accept ICMPv6 type 128 code 0 packets but drop ICMPv6 type 128 code 1 packets. • Using the sacpy tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the ICMPv6 packets are dropped or allowed according to the rules applied based on type. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. IPv4 Source address: • Configure a filter on the TOE to drop and accept traffic with specified IPv4 source addresses. • Using the 'acumen_firewall_vpn_tests' tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the packets with specified IPv4 source addresses are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. IPv4 Destination Address: • Configure a filter on the TOE to drop and accept traffic with specified IPv4 destination addresses. • Using the 'acumen_firewall_vpn_tests' tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the packets with specified IPv4 destination addresses are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. IPv4 Transport Layer Protocol: • Configure a filter on the TOE to drop IPv4 packets with UDP traffic and accept packets for TCP traffic. • Using the 'acumen_firewall_vpn_tests' tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the IPV4 packets with specified transport layer protocol are either dropped or allowed. • Verify via packet capture that the IPV4 packets with specified transport layer protocol are either dropped or allowed. IPv6 Source address: • Configure a filter on the TOE to drop and accept traffic with specified IPv6 source addresses. • Using the scapy tool generate and send traffic that matches the applied filter.

	• Verify via TOE logs that the packets with specified IPv6 source addresses are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. IPv6 Destination Address: • Configure a filter on the TOE to drop and accept traffic with specified IPv6 destination addresses. • Using the scapy tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the packets with specified IPv6 destination addresses are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. IPv6 Transport Layer Protocol: • Configure a filter on the TOE to drop IPv6 packets with UDP traffic and accept packets for TCP traffic. • Using the scapy tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the IPV6 packets with specified transport layer protocol are either dropped or allowed. • Verify via packet capture that the IPV6 packets with specified transport layer protocol are either dropped or allowed. TCP Source Port: • Configure a filter on the TOE to drop and accept traffic according to specified TCP source ports. • Using the 'acumen_firewall_vpn_tests' tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the TCP packets with specified source ports are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. TCP Destination Port: • Configure a filter on the TOE to drop and accept traffic according to specified TCP destination ports. • Using the 'acumen_firewall_vpn_tests' tool generate and send traffic that matches the applied filter. • Verify via TOE logs that the TCP packets with specified destination ports are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. UDP Source Port: • Configure a filter on the TOE to drop and accept traffic according to specified UDP source ports. • Generate and send traffic that matches the applied filter.

	• Verify via TOE logs that the UDP packets with specified source ports are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration. UDP Destination Port: • Configure a filter on the TOE to drop and accept traffic according to specified UDP destination ports. • Generate and send traffic that matches the applied filter. • Verify via TOE logs that the UDP packets with specified destination ports are either dropped or allowed. • Verify via packet capture that the appropriate traffic was allowed or denied as per the configuration.
Pass/Fail with Explanation	Pass. For ICMPv4, ICMPv6, TCP, UDP, IPv4 and IPv6, TOE successfully implemented full packet filter firewall rules that permit, drop, and log packets for each of the specified attributes.

6.1.10.4 FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4 TEST #2 [TD0924] (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	Test 2: Repeat the test assurance activity above to ensure that state full traffic filtering rules can be defined for each distinct network interface type supported by the TOE. Note that these test activities should be performed in conjunction with those of FFW_RUL_EXT.1.9 where the effectiveness of the rules is tested. The test activities for FFW_RUL_EXT.1.9 define the protocol/attribute combinations required to be tested. If those combinations are configured manually, that will fulfil the objective of these test activities, but if those combinations are configured otherwise (e.g., using automation), these test activities may be necessary in order to ensure the guidance is correct and the full range of configurations can be achieved by a TOE administrator. **TD0924 has been applied.**
Pass/Fail with Explanation	Pass. This test requirement has been covered as part of FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4 Test #1 (The TOE consists of only one interface type).

6.1.10.5 FFW_RUL_EXT.1.5 TEST #1 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	The following tests shall be run using IPv4 and IPv6.

	Test 1: The evaluator shall configure the TOE to permit and log TCP traffic. The evaluator shall initiate a TCP session. While the TCP session is being established, the evaluator shall introduce session establishment packets with incorrect flags to determine that the altered traffic is not accepted as part of the session (i.e., a log event is generated to show the ruleset was applied). After a TCP session is successfully established, the evaluator shall alter each of the session determining attributes (source and destination addresses, source and destination ports, sequence number, flags) one at a time in order to verify that the altered packets are not accepted as part of the established session.

Test Steps	IPv4:
	• Configure firewall rules on the TOE to permit and log TCP traffic but drop and log invalid TCP packets.
	• Start a listening on the destination VM (LAN) using netcat.
	• Develop and run a script to introduce session establishment packets with incorrect flags during session establishment followed by altered session determining attributes (source and destination addresses, source and destination ports, sequence number, flags) which are not accepted as part of the session.
	• Verify via TOE logs that the valid and altered TCP traffic is recorded by the firewall filter.
	  ○ Valid TCP SYN
	  ○ Invalid TCP flags before session establishment
	  ○ Altered source address
	  ○ Altered destination address
	  ○ Altered source port
	  ○ Altered destination port
	  ○ Altered sequence number
	  ○ Altered TCP flags after session establishment
	• Verify via packet capture only the TCP connection with valid flags is established and the altered traffic is not accepted as part of the session.
	  ○ Valid TCP SYN
	  ○ Invalid TCP flags
	  ○ Source address
	  ○ Destination address
	  ○ Source port
	  ○ Destination port
	  ○ Sequence number
	  ○ Flag
	
	IPv6:
	• Configure firewall rules on the TOE to permit and log TCP traffic but drop and log invalid TCP packets.
	• Start a listening on the destination VM (LAN) using netcat.
	• Develop and run a script to introduce session establishment packets with incorrect flags during session establishment followed by altered session determining attributes (source and destination addresses, source and

	destination ports, sequence number, flags) which are not accepted as part of the session.
	• Verify via TOE logs that the valid and altered TCP traffic is recorded by the firewall filter.
	  ○ Valid TCP SYN
	  ○ Invalid TCP flags before session establishment
	  ○ Altered source address
	  ○ Altered destination address
	  ○ Altered source port
	  ○ Altered destination port
	  ○ Altered sequence number
	  ○ Altered TCP flags after session establishment
	• Verify via packet capture only the TCP connection with valid flags is established and the altered traffic is not accepted as part of the session.
	  ○ Valid TCP SYN
	  ○ Invalid TCP flags
	  ○ Source address
	  ○ Destination address
	  ○ Source port
	  ○ Destination port
	  ○ Sequence number
	  ○ Flag

Pass/Fail with Explanation	Pass. The evaluator verified that altered packets for IPv4 and IPv6 (source and destination addresses, source and destination ports, sequence number, and flags) are not accepted as a part of the established session after a TCP session is successfully established. This meets testing requirements.

6.1.10.6 FFW_RUL_EXT.1.5 TEST #2 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	The following tests shall be run using IPv4 and IPv6. Test 2: The evaluator shall terminate the TCP session established per Test 1 as described in the TSS. The evaluator shall then immediately send a packet matching the former session definition in order to ensure it is not forwarded through the TOE without being subject to the ruleset.
Pass/Fail with Explanation	Pass. Tested in conjunction with FFW_RUL_EXT.1.5 Test #1.

6.1.10.7 FFW_RUL_EXT.1.5 TEST #3 (MOD_CPP_FW_V1.4E)

Item	Data

Test Assurance Activity	The following tests shall be run using IPv4 and IPv6. Test 3: The evaluator shall expire (i.e., reach timeout) the TCP session established per Test 1 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.
Test Steps	IPv4: • Configure the TCP session expiration value of 60 seconds on the TOE. • Using the netcat utility, start a port listening on the destination VM (LAN VM). • Using the netcat utility establish a TCP session and wait for the session to expire in 60 seconds. • Verify the TCP traffic being received at the destination. • Once the session expires, send a packet that matches the former TCP session using the scapy tool. • Verify that no traffic is received on the destination VM. • Verify via the firewall logs the TCP packet similar to the former session being dropped. • Verify via packet capture that the TCP traffic matching the former TCP session is sent by the source VM but not received at the destination. IPv6: • Configure the TCP session expiration value of 60 seconds on the TOE. • Using the netcat utility, start a port listening on the destination VM (LAN VM). • Using the netcat utility establish a TCP session and wait for the session to expire in 60 seconds. • Verify the TCP traffic being received at the destination. • Once the session expires, send a packet that matches the former TCP session using the scapy tool. • Verify that no traffic is received on the destination VM. • Verify via the firewall logs the TCP packet similar to the former session being dropped. • Verify via packet capture that the TCP traffic matching the former TCP session is sent by the source VM but not received at the destination.
Pass/Fail with Explanation	Pass. Any TCP packet matching the former expired session for both IPv4 and IPv6 is not forwarded through the TOE without being subject to the ruleset. This meets the testing requirements.

6.1.10.8 FFW_RUL_EXT.1.5 TEST #4 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	The following tests shall be run using IPv4 and IPv6.

	Test 4: The evaluator shall configure the TOE to permit and log UDP traffic. The evaluator shall establish a UDP session. Once a UDP session is established, the evaluator shall alter each of the session determining attributes (source and destination addresses, source and destination ports) one at a time in order to verify that the altered packets are not accepted as part of the established session.
Test Steps	IPv4: • Configure a firewall filter on the TOE to permit and log valid UDP sessions while dropping and logging altered packets. • Using the netcat tool, start a UDP port listening on the destination VM. • Using the 'acumen_firewall_vpn_tests' tool, modify and send each of the session attributes one at a time. • Verify the traffic being received for the valid connection on the destination VM. • Verify via TOE logs that the valid and the altered session traffic is recorded by the firewall filter. • Verify via packet capture that the valid and the altered session traffic is recorded by the firewall filter. IPv6: • Configure a firewall filter on the TOE to permit and log valid UDP sessions while dropping and logging altered packets. • Using the netcat tool, start a UDP port listening on the destination VM. • Using the 'acumen_firewall_vpn_tests' tool, modify and send each of the session attributes one at a time. • Verify the traffic being received for the valid connection on the destination VM. • Verify via TOE logs that the valid and the altered session traffic is recorded by the firewall filter. • Verify via packet capture that the valid and the altered session traffic is recorded by the firewall filter.
Pass/Fail with Explanation	Pass. The TOE does not accept altered packets (source and destination addresses, source and destination ports) as part of the established session after a UDP session is successfully established for both IPv4 and IPv6. This meets the testing requirement.

6.1.10.9 FFW_RUL_EXT.1.5 TEST #5 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	The following tests shall be run using IPv4 and IPv6. Test 5: The evaluator shall expire (i.e., reach timeout) the UDP session established per Test 4 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

Test Steps	IPv4: • Configure the UDP inactivity timeout values of 60 seconds on the TOE. • Using the netcat utility, start a port listening on the destination VM (LAN VM). • Using the netcat utility establish a UDP session and wait for the timeout (60 seconds). • Once the timeout is crossed, send a packet that matches the former UDP session using the scapy tool. • Verify via TOE logs that the UDP packet similar to the former session are recorded by the firewall logs. • Verify via packet capture that the UDP packet being sent is similar to the former session. IPv6: • Configure the UDP inactivity timeout values of 60 seconds on the TOE. • Using the netcat utility, start a port listening on the destination VM (LAN VM). • Using the netcat utility establish a UDP session and wait for the timeout (60 seconds). • Once the timeout is crossed, send a packet that matches the former UDP session using the scapy tool. • Verify via TOE logs that the UDP packet similar to the former session are recorded by the firewall logs. • Verify via packet capture that the UDP packet being sent is similar to the former session.
Pass/Fail with Explanation	Pass. The TOE does not forward packets matching a former expired UDP session for both IPv4 and IPv6 without subjecting them to the ruleset. This meets the testing requirement.

6.1.10.10 FFW_RUL_EXT.1.5 TEST #6 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	The following tests shall be run using IPv4 and IPv6. Test 6: If ICMP is selected, the evaluator shall configure the TOE to permit and log ICMP traffic. The evaluator shall establish a session for ICMP as defined in the TSS. Once an ICMP session is established, the evaluator shall alter each of the session determining attributes (source and destination addresses, other attributes chosen in FFW_RUL_EXT.1.5) one at a time in order to verify that the altered packets are not accepted as part of the established session.
Pass/Fail with Explanation	The support of ICMP is not claimed in the ST. Hence, this test is not applicable.

6.1.10.11 FFW_RUL_EXT.1.5 TEST #7 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	The following tests shall be run using IPv4 and IPv6. Test 7: If applicable, the evaluator shall terminate the ICMP session established per Test 6 as described in the TSS. The evaluator shall then immediately send a packet matching the former session definition in order to ensure it is not forwarded through the TOE without being subject to the ruleset.
Pass/Fail with Explanation	The support of ICMP is not claimed in the ST. Hence, this test is not applicable.

6.1.10.12 FFW_RUL_EXT.1.5 TEST #8 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	The following tests shall be run using IPv4 and IPv6. Test 8: The evaluator shall expire (i.e., reach timeout) the ICMP session established per Test 6 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.
Pass/Fail with Explanation	The support of ICMP is not claimed in the ST. Hence, this test is not applicable.

6.1.10.13 FFW_RUL_EXT.1.6 TEST #1 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	Both IPv4 and IPv6 shall be tested for items a), b), c), d), and e) of the SFR element. Both IPv4 and IPv6 shall be tested for item i) unless the rule definition is specific to IPv4 or IPv6. Note: f), g), and h) are specific to IPv4 or IPv6 and shall be tested accordingly Test 1: The evaluator shall test each of the conditions for automatic packet rejection in turn. In each case, the TOE should be configured to allow all network traffic and the evaluator shall generate a packet or packet fragment that is to be rejected. The evaluator shall use packet captures to ensure that the unallowable packet or packet fragment is not passed through the TOE.
Test Steps	IPv4: • Using the 'acumen-firewall-vpn' tool, send packets that meet testing conditions: a. Packets which are invalid fragments.

	b. Fragments that cannot be completely re-assembled. c. Packets where the source address is defined as being on a broadcast network. d. Packets where the source address is defined as being on a multicast network. e. Packets where the source address is defined as being a loopback address. f. Packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address "reserved for future use" (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4. • Using hping3, send packets with the IP options for Loose Source Routing, Strict Source Routing, and Record Routing. • Verify the invalid packets being sent via packet capture. • Verify via TOE logs, the counter being incremented for invalid IP fragments. • Verify via TOE logs, the counter being incremented for fragmented packets which cannot be re-assembled completely. • Verify via TOE logs, the packets with invalid source and destination addresses are dropped. • Verify via TOE logs the packets with loose, strict and record routing options are dropped. IPv6: • Configure a firewall rule to keep track of the fragmented packets being received. • Using the 'acumen-firewall-vpn' tool, send packets that meet testing conditions: a. Packets which are invalid fragments. b. Fragments that cannot be completely re-assembled. c. Packets where the source address is defined as being on a broadcast network. d. Packets where the source address is defined as being on a multicast network. e. Packets where the source address is defined as being a loopback address. f. Packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6. • Verify the invalid packets being sent via packet capture. • Verify via TOE logs, the counter being incremented for invalid IP fragments. • Verify via TOE logs, the counter being incremented for fragmented packets which cannot be re-assembled completely. • Verify via TOE logs, the packets with invalid source and destination addresses are dropped.
Pass/Fail with Explanation	Pass. For IPv4 and IPv6 each of the conditions (invalid fragment, fragments that cannot be reassembled, broadcast network source address, multicast network

	source address, loopback address, unspecified or reserved address, and packets with IPv4 options) are rejected and logged. This meets the testing requirements.

6.1.10.14 FFW_RUL_EXT.1.6 TEST #2 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	Both IPv4 and IPv6 shall be tested for items a), b), c), d), and e) of the SFR element. Both IPv4 and IPv6 shall be tested for item i) unless the rule definition is specific to IPv4 or IPv6. Note: f), g), and h) are specific to IPv4 or IPv6 and shall be tested accordingly Test 2: For each of the cases above, the evaluator shall use any applicable guidance to enable dropped packet logging or counting. In each case above, the evaluator shall ensure that the rejected packet or packet fragment was recorded (either logged or an appropriate counter incremented).
Pass/Fail with Explanation	Pass. The requirements of this test have been completed as part of testing for FFW_RUL_EXT.1.6 Test #1.

6.1.10.15 FFW_RUL_EXT.1.7 TEST #1 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	The following tests shall be run using IPv4 and IPv6. Test 1: The evaluator shall configure the TOE to drop and log network traffic where the source address of the packet matches that of the TOE network interface upon which the traffic was received. The evaluator shall generate suitable network traffic to match the configured rule and verify that the traffic is dropped, and a log message generated.
Test Steps	Note: The TOE does not require any firewall filters to be configured for this test in case of IPv4. Spoof checking is done by default by the TOE kernel. IPv4: • Verify the IP address configured for the WAN interface of the TOE. • Using the scapy tool, generate and send traffic that matches the created filter. • Verify via TOE logs that the traffic was denied. • Verify via packet capture that the traffic was denied. IPv6: • Verify the IP address configured for the WAN interface of the TOE. • Configure the TOE to drop and log network traffic where the source address of the packet matches that of the TOE network interface address. • Using the scapy tool, generate and send traffic that matches the created filter.

	• Verify via TOE logs that the traffic was denied. • Verify via packet capture that the traffic was denied.
Pass/Fail with Explanation	Pass. The TOE drops and logs the network traffic for both IPv4 and IPv6 when the source address of the packet matches the network interface upon which the traffic was received. This meets the testing requirement.

6.1.10.16 FFW_RUL_EXT.1.7 TEST #2 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	The following tests shall be run using IPv4 and IPv6. Test 2: The evaluator shall configure the TOE to drop and log network traffic where the source IP address of the packet fails to match the network reachability information of the interface to which it is targeted, e.g. if the TOE believes that network 192.168.1.0/24 is reachable through interface 2, network traffic with a source address from the 192.168.1.0/24 network should be generated and sent to an interface other than interface 2. The evaluator shall verify that the network traffic is dropped, and a log message generated.
Test Steps	IPv4: • Configure a filter to drop and log network traffic when the source IP address of the packet does not match the network reachability information of the TOE interface and accept otherwise. • Using the scapy tool, first the send traffic that matches the network reachability information of the TOE interface and then send the traffic that does not match the network reachability information of the TOE interface. • Verify via TOE logs that the traffic matching the TOE's network reachability information was accepted and the traffic that did not match the TOE's network reachability information was dropped. • Verify via a packet capture that the traffic matching the TOE's network reachability information was accepted and the traffic that did not match the TOE's network reachability information was dropped. IPv6: • Configure a filter to drop and log network traffic when the source IP address of the packet does not match the network reachability information of the TOE interface and accept otherwise. • Using the scapy tool, first the send traffic that matches the network reachability information of the TOE interface and then send the traffic that does not match the network reachability information of the TOE interface. • Verify via TOE logs that the traffic matching the TOE's network reachability information was accepted and the traffic that did not match the TOE's network reachability information was dropped.

	• Verify via a packet capture that the traffic matching the TOE's network reachability information was accepted and the traffic that did not match the TOE's network reachability information was dropped.
Pass/Fail with Explanation	Pass. The TOE drops and logs network traffic for both IPv4 and IPv6 when the source address of the packet fails to match the network reachability information of the interface upon which it was received. This meets the testing requirement.

6.1.10.17 FFW_RUL_EXT.1.8 TEST #1 [TD0545] (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	Test 1: If the TOE implements a mechanism that ensures that no conflicting rules can be configured, the evaluator shall try to configure two conflicting rules and verify that the TOE rejects the conflicting rule(s). It is important to verify that the mechanism is implemented in the TOE but not in the non-TOE environment. If the TOE does not implement a mechanism that ensures that no conflicting rules can be configured, the evaluator shall devise two equal stateful traffic filtering rules with alternate operations – permit and drop. The rules should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first rule is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation. **TD0545 has been applied.**
Test Steps	Note: TOE does not implement a mechanism that ensures that no conflicting rules can be configured. When conflicting rules are present in the firewall filter, the TOE evaluates them sequentially, and the first matching rule dictates the action taken on the packet. IPv4: • Configure a filter to allow and drop packets to a specific destination IP with the allow rule taking precedence. • Using scapy tool, send the traffic to the configured destination address in the filter. • Verify via TOE logs that traffic to the destination IP is being allowed. • Verify via packet capture that traffic to the destination IP is being allowed. • Configure a filter to allow and drop packets to a specific destination IP with the deny rule taking precedence. • Using scapy tool, send the traffic to the configured destination address in the filter. • Verify via TOE logs that traffic to the destination IP is being denied. • Verify via packet capture that traffic to the destination IP is being denied. IPv6:

	• Configure a filter to allow and drop packets to a specific destination IP with the allow rule taking precedence. • Using scapy tool, send the traffic to the configured destination address in the filter. • Verify via TOE logs that traffic to the destination IP is being allowed. • Verify via packet capture that traffic to the destination IP is being allowed. • Configure a filter to allow and drop packets to a specific destination IP with the deny rule taking precedence. • Using scapy tool, send the traffic to the configured destination address in the filter. • Verify via TOE logs that traffic to the destination IP is being denied. • Verify via packet capture that traffic to the destination IP is being denied.
Pass/Fail with Explanation	Pass. The TOE enforces the first rule for both IPv4 and IPv6 when two equal stateful traffic filtering rules with conflicting operations (permit and drop) are configured and deployed in two distinct orders. This meets the testing requirement.

6.1.10.18 FFW_RUL_EXT.1.8 TEST #2 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	Test 2: The evaluator shall repeat the procedure above, except that the two rules should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.
Test Steps	IPv4: • Configure the firewall rule so that the first rule allows packets to a specific destination address and the second rule denies packets to its network segment with the allow rule taking precedence. • Using the scapy tool, send the traffic to the configured destination address in the filter. • Verify via TOE logs that traffic to the destination address is being allowed. • Verify via packet capture that traffic to the destination address is being allowed. • Configure the firewall rule so that the first rule denies packets to a network segment and the second rule allows packets to a specific destination address of the network segment with the deny rule taking precedence. • Using the scapy tool, send the traffic to a destination address from the configured network segment in the filter. • Verify via TOE logs that traffic to the destination address is being denied. • Verify via packet capture that traffic to the destination address is being denied.

	IPv6: • Configure the firewall rule so that the first rule allows packets to a specific destination address and the second rule denies packets to its network segment with the allow rule taking precedence. • Using the scapy tool, send the traffic to the configured destination address in the filter. • Verify via TOE logs that traffic to the destination address is being allowed. • Verify via packet capture that traffic to the destination address is being allowed. • Configure the firewall rule so that the first rule denies packets to a network segment and the second rule allows packets to a specific destination address of the network segment with the deny rule taking precedence. • Using the scapy tool, send the traffic to a destination address from the configured network segment in the filter. • Verify via TOE logs that traffic to the destination address is being denied. • Verify via packet capture that traffic to the destination address is being denied.
Pass/Fail with Explanation	Pass. Regardless of the rule's specificity, the TOE enforces the first rule for both IPv4 and IPv6 when two rules are devised where one is a subset of the other (e.g., a specific address vs. a network segment).

6.1.10.19 FFW_RUL_EXT.1.9 TEST #1 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	For each attribute in FFW_RUL_EXT.1.2, the evaluator shall construct a test to demonstrate that the TOE can correctly compare the attribute from the packet header to the ruleset, and shall demonstrate both the permit and deny for each case. It shall also be verified that a packet is dropped if no matching rule can be identified for the packet. The evaluator shall check the log in each case to confirm that the relevant rule was applied. The evaluator shall record a packet capture for each test to demonstrate the correct TOE behaviour.
Pass/Fail with Explanation	Pass. This test has been completed as part of FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4 TEST #1.

6.1.10.20 FFW_RUL_EXT.1.10 TEST #1 (MOD_CPP_FW_V1.4E)

Item	Data
Test Assurance Activity	The following tests shall be run using IPv4 and IPv6.

	Test 1: The evaluator shall define a TCP half-open connection limit on the TOE. The evaluator shall generate TCP SYN requests to pass through the TOE to the target system using a randomised source IP address and common destination IP address. The number of SYN requests should exceed the TCP half-open threshold defined on the TOE. TCP SYN-ACK messages should not be acknowledged. The evaluator shall verify through packet capture that once the defined TCP half-open threshold has been reached, subsequent TCP SYN packets are not transmitted to the target system. The evaluator shall verify that when the configured threshold is reached that, depending upon the selection, either a log entry is generated or a counter is incremented.
Test Steps	IPv4: • Configure the firewall rule on the TOE to limit the amount of half-open TCP connections being received to a certain destination address. • Using scapy, form a script to generate the TCP SYN requests to pass through the TOE using a randomized source IP address and common destination IP address. • Run the script and send the traffic through the TOE to the destination address. • Verify via TOE logs that the TCP SYN packets are only allowed to the configure limit and are dropped once the threshold is crossed. • Verify via packet capture that the TOE drops the half-open TCP connections after the configured threshold is crossed. IPv6: • Configure the firewall rule on the TOE to limit the amount of half-open TCP connections being received to a certain destination address. • Using scapy, form a script to generate the TCP SYN requests to pass through the TOE using a randomized source IP address and common destination IP address. • Run the script and send the traffic through the TOE to the destination address. • Verify via TOE logs that the TCP SYN packets are only allowed to the configure limit and are dropped once the threshold is crossed. • Verify via packet capture that the TOE drops the half-open TCP connections after the configured threshold is crossed.
Pass/Fail with Explanation	Pass. The TOE detects when the number of SYN requests exceeds the TCP half-open threshold for both IPv4 and IPv6 and denies traffic once the counter has been reached. This meets the testing requirements.

7 CAVP TABLE MAPPING

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

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #	TOE Cards
FCS_CKM.1	ECC schemes using 'NIST curves' [**P-384, P-521**] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4, or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6;	CACI idt GoSilent Crypto Library v1.0	ECDSA KeyGen (FIPS186-4) ECDSA KeyVer (FIPS186-4) Curve: P-384, P-521	A7437	Debian 12 on Red Hat RHEL v9.6 KVM on Intel Xeon Silver 4514Y (Raptor Cove) Debian 12 on VMware ESXi v8.0 on Intel Xeon Silver 4514Y (Raptor Cove) Armbian 25.05 on AllWinner H5/Cortex A-53 (ARM v8-A)
FCS_CKM.1/IKE	FIPS PUB 186-5, "Digital Signature Standard (DSS)," Appendix B.4 for ECDSA schemes, and implementing "NIST curves" P-384 and [**P-521**]	CACI idt GoSilent Crypto Library v1.0	ECDSA KeyGen (FIPS186-5) Curve: P-384, P-521	A7437	Debian 12 on Red Hat RHEL v9.6 KVM on Intel Xeon Silver 4514Y (Raptor Cove) Debian 12 on VMware ESXi v8.0 on Intel Xeon Silver 4514Y (Raptor Cove) Armbian 25.05 on AllWinner H5/Cortex A-53 (ARM v8-A)
FCS_CKM.2	Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	CACI idt GoSilent Crypto Library v1.0	KAS_ECC_CDH-Component SP800-56Ar3 Curve: P-384, P-521	A7437	Debian 12 on Red Hat RHEL v9.6 KVM on Intel Xeon Silver 4514Y (Raptor Cove) Debian 12 on VMware ESXi v8.0 on Intel Xeon Silver 4514Y (Raptor Cove) Armbian 25.05 on AllWinner H5/Cortex A-53 (ARM v8-A)
FCS_COP.1/ DataEncryption	AES used in [**GCM**] mode and cryptographic key sizes [**256 bits**]	CACI idt GoSilent Crypto Library v1.0	AES-GCM Direction: Decrypt, Encrypt Key Length: 256	A7437	Debian 12 on Red Hat RHEL v9.6 KVM on Intel Xeon Silver 4514Y (Raptor Cove) Debian 12 on VMware ESXi

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #	TOE Cards
			AES: ISO 18033-3 GCM: ISO 19772		v8.0 on Intel Xeon Silver 4514Y (Raptor Cove)
					Armbian 25.05 on AllWinner H5/Cortex A-53 (ARM v8-A)
FCS_COP.1/ SigGen	For ECDSA schemes implementing [**P-384, P-521**] curves that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST Recommended" curves; or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.	CACI idt GoSilent Crypto Library v1.0	ECDSA SigGen (FIPS186-4) ECDSA SigVer (FIPS186-4) Curve: P-384, P-521	A7437	Debian 12 on Red Hat RHEL v9.6 KVM on Intel Xeon Silver 4514Y (Raptor Cove)
					Debian 12 on VMware ESXi v8.0 on Intel Xeon Silver 4514Y (Raptor Cove)
					Armbian 25.05 on AllWinner H5/Cortex A-53 (ARM v8-A)
FCS_COP.1/ Hash	[**SHA-256, SHA-384, SHA-512**] and message digest sizes [**256, 384, 512**] bits	CACI idt GoSilent Crypto Library v1.0	SHA-256, SHA-384, SHA-512 Message Length: 0-65536 Increment 8 ISO/IEC 10118-3:2004	A7437	Debian 12 on Red Hat RHEL v9.6 KVM on Intel Xeon Silver 4514Y (Raptor Cove)
					Debian 12 on VMware ESXi v8.0 on Intel Xeon Silver 4514Y (Raptor Cove)
					Armbian 25.05 on AllWinner H5/Cortex A-53 (ARM v8-A)
FCS_COP.1/ KeyedHash	[**HMAC-SHA-384, HMAC-SHA-512**] and cryptographic key sizes [**key size (in bits) used in HMAC**] and message digest sizes [**384, 512**] bits	CACI idt GoSilent Crypto Library v1.0	HMAC-SHA2-384, HMAC-SHA2-512 MAC: 384, 512 Key Length: 384, 512 ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"	A7437	Debian 12 on Red Hat RHEL v9.6 KVM on Intel Xeon Silver 4514Y (Raptor Cove)
					Debian 12 on VMware ESXi v8.0 on Intel Xeon Silver 4514Y (Raptor Cove)
					Armbian 25.05 on AllWinner H5/Cortex A-53 (ARM v8-A)

SFR	Algorithm Description	Implementation name	CAVP Alg.	CAVP Cert #	TOE Cards
FCS_RBG_EXT.1	**CTR_DRBG (AES)**	CACI idt GoSilent Crypto Library v1.0	Counter DRBG Capabilities: Mode: AES-256 Derivation Function Enabled: Yes Additional Input: 0-256 Increment 256 Entropy Input: 256-512 Increment 128 Nonce: 128 Personalization String Length: 0-256 Increment 256 Returned Bits: 256	A7437	Debian 12 on Red Hat RHEL v9.6 KVM on Intel Xeon Silver 4514Y (Raptor Cove) Debian 12 on VMware ESXi v8.0 on Intel Xeon Silver 4514Y (Raptor Cove) Armbian 25.05 on AllWinner H5/Cortex A-53 (ARM v8-A)

8 SFR DISTRIBUTION BETWEEN COMPONENTS

The following table addresses the SFR distribution requirements between TOE components, as required in Section 3.1 of the Firewall Module. Since SFRs drawn from the VPN Gateway Module are not addressed by the distribution requirements in the Firewall Module, their distribution requirements in the following table have been specified to be consistent with the requirements.

Table 1: SFR Distribution Between Components

SFR	Dist. Requirement	GoSilent Server	GoSilent Cube
FAU_GEN.1	All	X	X
FAU_GEN.1/VPN	All	X	X
FAU_GEN.2	All	X	X
FAU_GEN_EXT.1	All	X	X
FAU_STG_EXT.1	All	X	X
FAU_STG_EXT.4	Feature Dependent	X	X
FCO_CPC_EXT.1	All	X	X
FCS_CKM.1	One	X	X
FCS_CKM.1/IKE	One	X	X
FCS_CKM.2	All	X	X

SFR	Dist. Requirement	GoSilent Server	GoSilent Cube
FCS_CKM.4	All	X	X
FCS_COP.1/DataEncryption	All	X	X
FCS_COP.1/SigGen	All	X	X
FCS_COP.1/Hash	All	X	X
FCS_COP.1/KeyedHash	All	X	X
FCS_HTTPS_EXT.1	Feature Dependent	X	X
FCS_IPSEC_EXT.1	Feature Dependent	X	X
FCS_RBG_EXT.1	All	X	X
FCS_TLSC_EXT.1	Feature Dependent	X	X
FCS_TLSS_EXT.1	Feature Dependent	X	X
FDP_RIP.2	Feature Dependent	X	X
FFW_RUL_EXT.1	One	X	
FIA_AFL.1	One	X	X
FIA_PMG_EXT.1	One	X	X
FIA_PSK_EXT.1	Feature Dependent	X	X
FIA_PSK_EXT.2	Feature Dependent	X	X
FIA_UIA_EXT.1	One	X	X
FIA_UAU.7	Feature Dependent	X	X
FIA_X509_EXT.1/ITT	Feature Dependent	X	X
FIA_X509_EXT.1/Rev	Feature Dependent	X	X
FIA_X509_EXT.2	Feature Dependent	X	X
FIA_X509_EXT.3	Feature Dependent	X	X
FMT_MOF.1/ManualUpdate	All	X	X
FMT_MOF.1/Services	Feature Dependent	X	X
FMT_MTD.1/CoreData	All	X	X
FMT_MTD.1/CryptoKeys	Feature Dependent	X	X
FMT_SMF.1	Feature Dependent	X	X
FMT_SMF.1/FFW	Feature Dependent	X	
FMT_SMF.1/VPN	Feature Dependent	X	
FMT_SMR.2	One	X	X
FPF_RUL_EXT.1	Feature Dependent	X	
FPT_APW_EXT.1	Feature Dependent	X	X
FPT_FLS.1/SelfTest	All	X	X
FPT_ITT.1	Feature Dependent	X	X
FPT_SKP_EXT.1	All	X	X
FPT_STM_EXT.1	All	X	X
FPT_TST_EXT.1	All	X	X

SFR	Dist. Requirement	GoSilent Server	GoSilent Cube
FPT_TST_EXT.3	All	X	X
FPT_TUD_EXT.1	All	X	X
FTA_SSL_EXT.1	Feature Dependent	X	X
FTA_SSL.3	Feature Dependent	X	X
FTA_SSL.4	Feature Dependent	X	X
FTA_TAB.1	One	X	X
FTP_ITC.1	One	X	X
FTP_ITC.1/VPN	One	X	X
FTP_TRP.1/Admin	One	X	X

9 CONCLUSION

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