

**Assurance Activity Report
For
SonicWALL SMA v12.4**

Version 0.7

09/22/2021

VID: 11218

Produced by:



7925 Jones Branch Dr. #5200, McLean, VA 22102

Prepared for:

**National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme**

The Developer of the TOE:

SonicWALL
1033 McCarthy Blvd,
Milpitas, CA 95035

The Security Target Developed By:

CygnaCom Solutions Inc.
7925 Jones Branch Dr. #5200,
McLean, VA 22102

The TOE Evaluation Sponsored By:

SonicWALL

SonicWALL SMA v12.4 Assurance Activity Report

Table of Contents

- 1 Introduction 6**
 - 1.1 References..... 6**
 - 1.2 Target of Evaluation..... 6**
 - 1.3 Platform Equivalence..... 7**
 - 1.4 TOE Architecture Description 7**
 - 1.5 Testing Environment..... 7**
- 2 Security Functional Requirements..... 8**
 - 2.1 Security Audit (FAU) 8**
 - 2.1.1 FAU_GEN.1 Audit Data Generation8
 - 2.1.2 FAU_GEN.2 Audit Data Generation31
 - 2.1.3 FAU_STG_EXT.1 Protected Audit Event Storage32
 - 2.2 Cryptographic Support (FCS)..... 36**
 - 2.2.1 FCS_CKM.1 Cryptographic Key Generation (for asymmetric keys)36
 - 2.2.2 FCS_CKM.2 Cryptographic Key Establishment.....39
 - 2.2.3 FCS_CKM.4 Cryptographic Key Destruction.....41
 - 2.2.4 FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/Decryption)
44
 - 2.2.5 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification) 45
 - 2.2.6 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm).....46
 - 2.2.7 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)47
 - 2.2.8 FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)48
 - 2.2.9 FCS_TLSC_EXT.1 TLS Client Protocol48
 - 2.2.10 FCS_TLSS_EXT.1 TLS Server Protocol56
 - 2.2.11 FCS_TLSS_EXT.2 Extended: TLS Server support for mutual authentication64
 - 2.3 Identification and Authentication (FIA)..... 68**
 - 2.3.1 FIA_AFL.1.1 Authentication Failure Management.....68
 - 2.3.2 FIA_PMG_EXT.1 Password Management69
 - 2.3.3 FIA_UIA_EXT.1 User Identification and Authentication71
 - 2.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism.....72
 - 2.3.5 FIA_UAU.7 Protected Authentication Feedback72
 - 2.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation.....73
 - 2.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication78
 - 2.3.8 FIA_X509_EXT.3 Extended: Certificate Requests79

2.4	Security Management (FMT)	80
2.4.1	FMT_MOF.1/ManualUpdate Management of Security Functions Behavior.....	80
2.4.2	FMT_MTD.1/CoreData Management of TSF Data.....	81
2.4.3	FMT_MTD.1/CryptoKeys Management of TSF Data.....	82
2.4.4	FMT_SMF.1 Specification of Management Functions.....	83
2.4.5	FMT_SMR.2 Restrictions on Security Roles.....	85
2.5	Protection of the TSF (FPT)	86
2.5.1	FPT_SKP_EXT.1 Protection of TSF Data (for reading all symmetric keys).....	86
2.5.2	FPT_APW_EXT.1 Protection of Administrator Passwords.....	87
2.5.3	FPT_TST_EXT.1 TSF Testing.....	87
2.5.4	FPT_TUD_EXT.1 Trusted Update.....	89
2.5.5	FPT_STM_EXT.1 TSF Reliable Time Stamps.....	92
2.6	TOE Access (FPT)	93
2.6.1	FTA_SSL_EXT.1 TSF-initiated Session Locking.....	93
2.6.2	FTA_SSL.3 TSF-initiated Termination.....	94
2.6.3	FTA_SSL.4 User-initiated Termination.....	95
2.6.4	FTA_TAB.1 Default TOE Access Banners.....	96
2.7	Trusted Path/Channels (FTP)	97
2.7.1	FTP_ITC.1 Inter-TSF Trusted Channel.....	97
2.7.2	FTP_TRP.1/Admin Trusted Path.....	99
3	Security Assurance Requirements	99
3.1	ASE: Security Target Evaluation	100
3.1.1	General ASE.....	100
3.2	ADV_FSP.1 Basic Functional Specification	100
3.2.1	Assurance Activities.....	100
3.3	AGD: Guidance Documents	101
3.3.1	AGD_OPE.1 Operational User Guidance.....	101
3.3.2	Preparative Procedures.....	102
3.4	ALC: Life-cycle Support	102
3.4.1	ALC_CMC.1 Labelling of the TOE.....	102
3.4.2	ALC_CMS.1 TOE CM coverage.....	102
3.5	ATE: Tests	103
3.5.1	ATE_IND.1 Independent Testing – Conformance.....	103
3.6	AVA: Vulnerability Assessment	103

SonicWALL SMA v12.4 Assurance Activity Report

3.6.1 AVA_VAN.1 Vulnerability Survey103

List of Figures

Figure 1: Physical Lab Test Setup – SonicWALL Test Environment.....8

List of Tables

Table 1: Guidance and Reference Documents6
Table 2: TOE Platforms and Devices6
Table 3: SMA physical appliances7
Table 4: SMA virtual appliances.....7
Table 5: Auditable Events10
Table 6: Audits of Administrative Commands15
Table 7: SonicWALL SMA v12.4 CSPs42

1 Introduction

This document summarizes the evaluation results of a specific Target of Evaluation (TOE), SonicWALL SMA v12.4, conforming to the *collaborative Protection Profile for Network Devices, Version 2.2e, 23 March 2020*, by listing the assurance activities and associated results as performed by the evaluators.

1.1 References

The following table provides the information needed to identify and to control the Security Target (ST), the Target of Evaluation (TOE), and other evidence used in this evaluation.

Table 1: Guidance and Reference Documents

Item	Identifier	Short Form
Security Target	SonicWall SMA v12.4 Security Target Version 0.5 Sep 22, 2021	[ST]
Common Criteria Publications	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017.	[CC]
Protection Profile	collaborative Protection Profile for Network Devices Version 2.2e, 23 March 2020 (NDcPP)	[NDcPP]
Supporting Document	Evaluation Activities for Network Device cPP, December 2019, Version 2.2	[SD]
Common Criteria Configuration Guide	SonicWall SMA v12.4, Common Criteria Configuration Guide, Version 1.1 July 2021	[CC-Addendum]
User Guidance	SonicWALL Secure Mobile Access 12.4 Administration Guide	[ADMIN]
Test Report	Test Report v0.5 SonicWALL SMA v12.4 September 13, 2021	[TSTRPT]
Audit Report	SonicWall SMA v12.4 Audit Events Report v0.3 July 27, 2021	[AUDITR]
Entropy Report	SonicWALL version 12.4.1 Testing Topology and Configuration version 0.3.docx	[ENT]

1.2 Target of Evaluation

The evaluated product name is SonicWall Secure Mobile Access (SMA) v12.4, and the evaluated version of the TOE is 12.4.1. The Target of Evaluation [TOE] is a Network Device as defined by the *collaborative Protection Profile for Network Devices v2.2e* [NDcPP]: “A network device in the context of this cPP is a device composed of both hardware and software that is connected to the network and has an infrastructure role within the network”.

The TOE, SonicWall SMA v12.4, is offered as physical appliances, which consists of SMA 6210 and SMA 7210 appliances and the SMA 8200v virtual appliance. The TOE consists of both hardware and software components. The SMA 6210 and SMA 7210 are identical except for CPU and 2 additional SFP+ network ports. The SMA 8200v is a virtual appliance designed to operate in the VMware Hypervisor version 6.7 virtualization environment.

All the physical TOE appliances are shipped ready for immediate access through a Command Line Interface (CLI) and after basic network configuration through a web-based Appliance Management Console (AMC). Virtual appliance requires installation into hypervisor environment and supports configuration through AMC. To ensure secure use the TOE must be configured prior to being put into production environment as specified in the user guidance.

The following table lists the TOE platform and Devices.

Table 2: TOE Platforms and Devices

Series	Platforms	Build
SonicWall Secure Mobile Access	SMA 6210	

SonicWALL SMA v12.4 Assurance Activity Report

SMA1000 Series	SMA 7210	12.4.1-02451 ¹
	SMA 8200v	

1.3 Platform Equivalence

The TOE consists of three appliances (SMA 6210, SMA 7210 and SMA8200v), all were fully tested. As a result of full coverage, the requirement to present equivalency argument is trivially satisfied.

1.4 TOE Architecture Description

The SonicWall Secure Mobile Access (SMA) v12.4 appliance functions as a remote access gateway operating as an intermediary device between end users on client devices and network resources residing on internal network. The appliance provides multiple access methods for end users or client devices to remotely access internal network resources from untrusted external networks. The SMA administrator configures policies comprised of security rules operating on users and targeting resources that must be satisfied in order to establish remote access.

1.5 Testing Environment

The TOE, SonicWall SMA v12.4, is offered as SMA 6210 and SMA 7210 hardware appliances and SMA 8200v virtual appliance. The TOE consists of both hardware and software components. The SMA 6210 and SMA 7210 are identical except for CPU, RAM, and SFP+ ports. The SMA 8200v is a virtual appliance designed to operate in the VMware Hypervisor version 6.7 virtualization environment. The following tables 2 and 3 summarize the compounds:

Table 3: SMA physical appliances

Platform	Model	Processor	Form	RAM	Specs	Build Number
SMA v12.4	SMA 6210	Intel Core i5-7500 (Kaby Lake)	1U	8GB (DDR4)	6 * 1GB Ports	12.4.1-02451 ²
	SMA 7210	Intel Xeon E3-1275 v6 (Kaby Lake)	1U	16GB(DDR4)	6 * 1GB, 2 * 10GB SFP+ Ports	12.4.1-02451 ³

Table 4: SMA virtual appliances

Platform	Model	Hypervisor	OS	CPU	RAM	Hard disk space	Virtual NIC	Hard disk space	Build Number
SMA v12.4	SMA 8200v	ESXi 6.7	SMA1000	4 vCPUs (Xeon Silver 4208 2.1GHz)	8GB ECC DDR- 4 2400	160 GB, thick provisioned	2 vNIC of 1000BaseT	250 GB, thick provisioned	12.4.1- 02451 ⁴

As shown below, the topology is configured for a dedicated and fully isolated 'Test' LAN. This setup prevents general access while still granting evaluators direct access to the TOE. The setup consists of a 'Test' LAN –

¹ Core build SMA 12.4.1-02451 with pform-hotfix-12.4.1-02559 and Connect Tunnel 12.4.1.939 was used in testing.

² Core build SMA 12.4.1-02451 with pform-hotfix-12.4.1-02559 and Connect Tunnel 12.4.1.939 was used in testing.

³ Core build SMA 12.4.1-02451 with pform-hotfix-12.4.1-02559 and Connect Tunnel 12.4.1.939 was used in testing.

⁴ Core build SMA 12.4.1-02451 with pform-hotfix-12.4.1-02559 and Connect Tunnel 12.4.1.939 was used in testing.

SonicWALL SMA v12.4 Assurance Activity Report

192.168.0.x/24 for IPv4. The server is local to the 'Test' LAN and packet capture is done by a laptop connected to a mirrored port on the switch. See Table 5 for details. All devices in the testing setup are synchronized through an NTP server virtual machine (IP:192.168.0.206/24, hostname= ntpd.lab.local).

Note: The diagram shows the components involved in the testing.

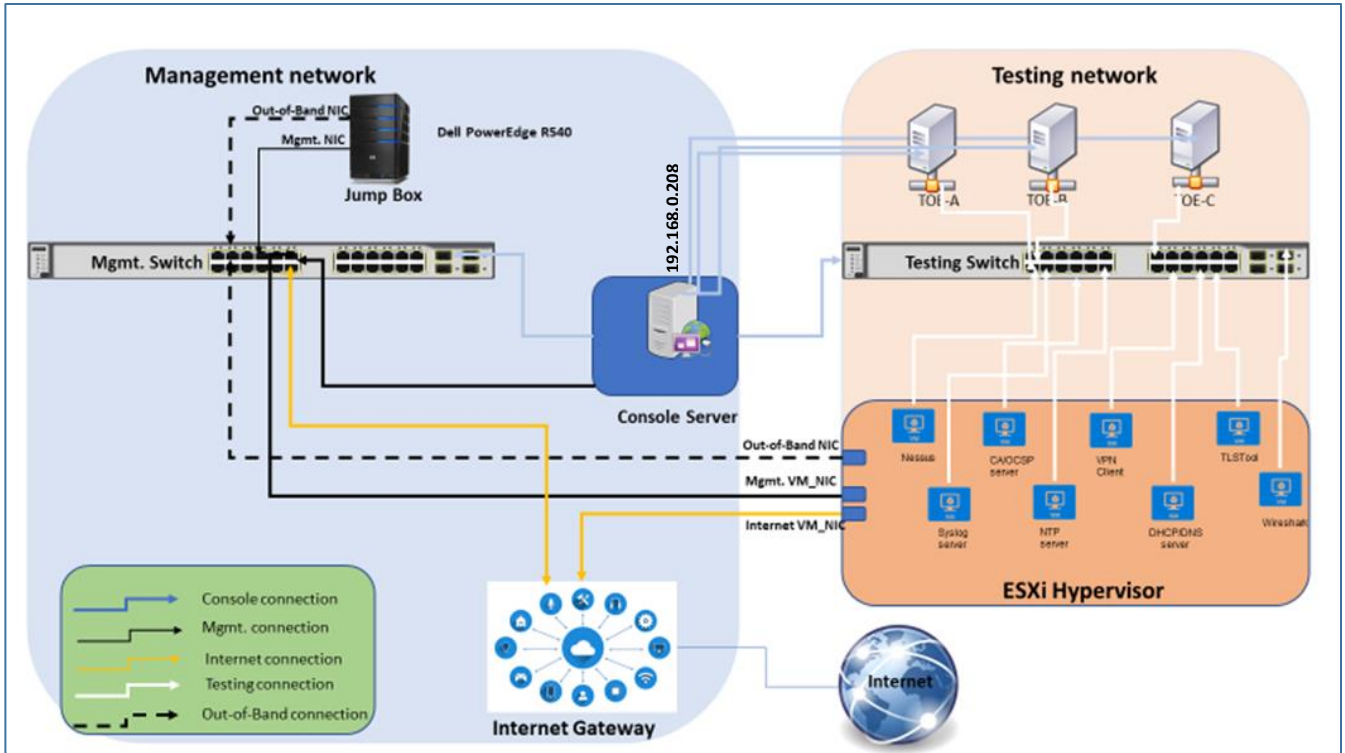


Figure 1: Physical Lab Test Setup – SonicWALL Test Environment

Please refer to TSTRPR for further details on the test environment.

2 Security Functional Requirements

2.1 Security Audit (FAU)

2.1.1 FAU_GEN.1 Audit Data Generation

2.1.1.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The evaluator confirmed that the ST Section 7.1 contains the categorical description of administrative actions related to cryptographic keys.

SonicWALL SMA v12.4 Assurance Activity Report

For the administrative task of generating/import of, changing, or deleting of cryptographic keys, following information are provided in the ST Section 7.1.

The TOE generates audit records for the following administrative tasks related to cryptographic keys:

- *Generation and destruction of a public and associated private key used to authenticate TOE's TLS server. Multiple key pairs can exist and identified in the audit records by CN.*
- *Installation and removal of a trusted root or intermediate authority certificate(s) are identified in the audit records by CN.*
- *Generation of CSR and import of a signed certificate used to authenticate TOE's TLS client. These are unique and identified in the audit records by CN.*

The evaluator confirmed that the TSS includes the following description as to what information is logged to identify keys: *"Multiple Key pairs can exist and identified in the audit records by Common Name (CN)"*.

Based on this information the evaluator concluded that the TSS adequately identifies relevant keys and describes the key identification scheme enabling authorized administrators to trace individual keys when reviewing the audit trail.

2.1.1.2 Guidance Assurance Activities

Guidance Assurance Activities:

- (1) 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).
- (2) The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes.
- (3) 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.
- (4) 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.

SonicWALL SMA v12.4 Assurance Activity Report

Guidance Implementation Details/Results:

- (1) The evaluator checked the CC-Addendum, Section 4 – Auditable Events and noted that it provides an example of each auditable event required by FAU_GEN.1. Since the format is same for all the auditable events, the CC-Addendum provides the global format for the events before the example table. The evaluator considers this description as adequate to both interpret the logs, which are human-readable, and to assist with searching for the specific records in the audit trail.
- (2) The evaluator checked the CC-Addendum and noted that it lists all of the TOE's audit event types. The evaluator crosschecked this list with the ST, Table 11 to ensure that every audit event mandated by the NDcPP is described. Each audit record contains the following information: type of event, date and time of the event, subject identity, and the outcome. All audit records contain this mandatory information.
- (3) The evaluator examined the CC-Addendum and made the determination that administrative commands fully meet the requirements outlines in the cPP. The evaluator determined that configuration of the TOE into the evaluated configuration generates appropriate level of audit records. The evaluator considered commands issued to the TOE as part of establishing evaluated configuration as well as management commands explicitly defined in the cPP to be security relevant.

Table 5: Auditable Events

Requirement (SFR)	Auditable Events	Additional Audit Record Contents	CC Guide Mapping
FAU_GEN.1	None.	None.	CC-Addendum, Section 4 Auditable Events
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).	<p><u>Unsuccessful administrative login:</u></p> <p>Warning 6/11/19 06:26:28 AMC Authentication failed: Username=admin, Address=10.1.101.10</p> <p><u>Unsuccessful login attempt limit is met or exceeded:</u></p> <p>Info 7/25/19 14:52:50 admin Added configuration extension - Key=ADMINISTRATOR_ACCOUNT_LOCKOUT_SECONDS Value=180 Info 7/25/19 14:52:50 admin Added configuration extension - Key=ADMINISTRATOR_ACCOUNT_LOCKOUT_ATTEMPTS Value=4 Error 8/5/19 11:58:13 admin Administrator account locked due to 3 successive login failures</p>
FCS_TLSS_EXT.1	Failure to establish a TLS session	Reason for failure	<p><u>SSL Handshake Failure</u></p> <p>Error 6/24/19 15:41:31 AMC SSL handshake failed: Client requested protocol TLSv1 not enabled or not supported.</p>

SonicWALL SMA v12.4 Assurance Activity Report

FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure	<p><u>SSL Handshake Failure</u></p> <p>Error 6/25/19 15:26:35 AMC SSL handshake failed: no cipher suites in common</p>
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).	<p><u>Successful administrative login:</u></p> <p>Info 6/11/19 09:00:14 admin Login succeeded - Address=10.1.101.10</p> <p><u>Unsuccessful administrative login:</u></p> <p>Warning 6/11/19 06:26:28 AMC Authentication failed: Username=admin, Address=10.1.101.10</p>
FIA_UAU_EXT.2	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).	<p><u>Successful administrative login:</u></p> <p>Info 6/11/19 09:00:14 admin Login succeeded - Address=10.1.101.10</p> <p><u>Unsuccessful administrative login:</u></p> <p>Warning 6/11/19 06:26:28 AMC Authentication failed: Username=admin, Address=10.1.101.10</p>
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate	Reason for failure of certificate validation	<p><u>Unsuccessful attempt to validate an X509 certificate</u></p> <p>Aug 8 18:56:24 syslog-ng@SMAAppliance syslog.err syslog-ng: Certificate subject does not match configured hostname; subject='/DC=com/DC=sma1000/CN=ROOT', hostname='10.1.111.101', certificate='ROOT'</p>
	Any addition, replacement or removal of trust anchors in the TOE's trust store	Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store	<p><u>Command to delete trusted CA</u></p> <p>Info 8/5/19 10:08:07 admin Deleted CA certificate - Issued to=Unit Testing CA</p> <p><u>Command to add trusted CA</u></p> <p>Info 8/8/19 09:25:19 admin Added CA certificate - Issued to=ROOTCA</p>

SonicWALL SMA v12.4 Assurance Activity Report

FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.	<u>Uploading a Valid hotfix file:</u> Info 6/24/19 10:47:57 admin Installed hotfix pform-hotfix-12.4.1-02559
FMT_SMF.1	All management activities of TSF data.	None.	<u>Configuring and modifying access banner</u> Info 8/2/19 17:57:08 admin Added configuration extension - Key=ACCEPTABLE_USE_BANNER Value=Welcome to AMC
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	None.	<u>Installing a Valid hotfix file:</u> Info 6/24/19 10:47:57 admin Installed hotfix pform-hotfix-12.4.1-02559 <u>Installing an Invalid hotfix file:</u> Error 8/2/19 17:36:15 admin Hotfix update failed: Hotfix file integrity check failed.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process.	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).	<u>Configuring System Time</u> Info 6/12/19 12:59:17 admin Set time to Wed Jun 12 12:59:17 IST 2019
FTA_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session locking mechanism.	None.	<u>Timeout of local administrative session</u> Sep 3 15:55:04 SMAAppliance -bash: Timeout, session closed for user(root) Sep 3 15:55:04 SMAAppliance login[4754]: pam_unix(login:session): session closed for user root
FTA_SSL.3	The termination of a remote session by the	None.	<u>Timeout of remote administrative session</u>

SonicWALL SMA v12.4 Assurance Activity Report


	session locking mechanism.		Logout - Address=192.168.56.1 Duration=03:15:57 Expired=true
FTA_SSL.4	The termination of an interactive session.	None.	<u>Administrator Log Off</u> Info 6/21/19 13:24:57 admin Logout - Address=10.5.22.125 Duration=00:00:26 Expired=false
FTP_ITC.1	Initiation of the trusted channel.	Identification of the initiator and target of failed trusted channels establishment attempt.	<u>Audit Server Successful Connection</u> Accepted syslog connection Aug 9 18:26:54 perfapp-224 syslog-ng[27222]: syslog-ng starting up; version='3.9.1' Aug 9 18:26:54 perfapp-224 syslog-ng[27222]: Syslog connection established; fd='17', server='AF_INET(192.168.0.204:6514)', local='AF_INET(0.0.0.0:0)'
	Termination of the trusted channel.		<u>Audit Server Unsuccessful Connection</u> Aug 9 18:26:54 perfapp-224 syslog-ng[27222]: Syslog connection failed; fd='17', server='AF_INET(192.168.0.204:6514)', local='AF_INET(0.0.0.0:0)'
	Failure of the trusted channel functions.		<u>Failure of the Trusted Channel</u> Aug 8 18:56:24 syslog-ng@SMAAppliance syslog.err syslog-ng: Certificate subject does not match configured hostname; subject='/DC=com/DC=sma1000/CN=ROOT', hostname='10.1.111.101', certificate='ROOT'
FTP_TRP.1/Admin	Initiation of the trusted path.	None.	<u>Example: Configure administrator account</u> Info 9/11/19 13:48:17 admin Added administrator account - Username= user1 Role= Super Admin <u>Successful Administrative Login:</u>

SonicWALL SMA v12.4 Assurance Activity Report


		Info 6/11/19 09:00:14 admin Login succeeded - Address=10.1.101.10
	Termination of the trusted path.	<u>Timeout of local administrative session</u> Logout - Address=192.168.56.1 Duration=03:15:57 Expired=true
	Failures of the trusted path functions.	<u>Unsuccessful login</u> Info 7/25/19 14:52:50 admin Added configuration extension - Key=ADMINISTRATOR_ACCOUNT_LOCKOUT_SECONDS Value=180 Info 7/25/19 14:52:50 admin Added configuration extension - Key=ADMINISTRATOR_ACCOUNT_LOCKOUT_ATTEMPTS Value=4 Error 8/5/19 11:58:13 admin Administrator account locked due to 3 successive login failures

- (4) During testing, the evaluator followed CC-Addendum, was **successful** in putting TOE into evaluated configuration, and noted that all explicitly defined (FMT_SMF.1) administrative actions listed. Therefore “administrative actions related to TSF data related to configuration changes” are present in the CC-Addendum. The evaluator examined the CC-Addendum, Section 4 and made the determination that administrative commands meet the requirements outlined in the cPP. The evaluator determined that configuration of the TOE into the evaluated configuration generates expected audit records. The evaluator considered commands issued to the TOE as part of establishing evaluated configuration as well as management commands explicitly defined in the cPP to be security relevant.

Table 6: Audits of Administrative Commands

Access Privilege	Administrative Actions	Commands Executed	Mapping to Guidance (CC-Addendum or ADMIN)
Administrator	Start-up and shut down of audit functions	<p>AMC:</p> <ul style="list-style-type: none"> Login to AMC. Navigate to Monitoring -> Logging -> View Logs -> Management Audit Log 	<p>ADMIN Section: System Logging and Monitoring; Page 299</p>
Administrator	Change of audit level	<p>AMC:</p> <ul style="list-style-type: none"> Login to AMC Navigate to Monitoring -> Logging -> Configure Logging Choose the log levels for the various services like: Error/Warning/Info/Verbose/Debug etc Click on "Save" button Navigate to Monitoring -> Logging -> View Logs -> Management Audit Log. 	<p>ADMIN Section: Configuring Log Settings; Page 296</p>
Administrator	Configure administrator roles	<p>AMC:</p> <ul style="list-style-type: none"> Login to AMC using administrative credentials Select System Configuration -> Authentication Servers Click New Select Local users under Local users storage from the right pane, and leave everything else unchanged  <ul style="list-style-type: none"> Click Continue Type "local-auth" in 	<p>CC-Addendum Section 3: Evaluated Configuration.</p> <p>Step#1 Create a new local authentication server and configure password policy; Page 21</p> <p>Step#2 Create a new local administrator; Page 23</p>

SonicWALL SMA v12.4 Assurance Activity Report

		<p>Name: * field</p> <ul style="list-style-type: none"> • Under Password policy checkbox Lowercase letters, Numeric digits (0-9), Uppercases letters, and Symbols check boxes • Click on Save button • Navigate to System Configuration → General Settings • Click on Authentication button • Select "local-auth" from the Authentication server: drop-down menu • Click on Save button • Apply Pending Changes • Select System Configuration → General Settings • Click Administrators • Click New and select Administrator... • Populate mandatory fields and click Save 	
<p>Administrator</p>	<p>Configure password complexity</p>	<p>AMC:</p> <ul style="list-style-type: none"> • Login to AMC using administrative credentials • Select System Configuration → Authentication Servers • Click New • Select Local users under Local users storage from the right pane, and leave everything else unchanged <p align="center">  </p> <ul style="list-style-type: none"> • Click Continue • Type "local-auth" in Name: * field • Under Password policy checkbox Lowercase letters, Numeric digits (0- 	<p>CC-Addendum Section 3: Evaluated Configuration;</p> <p>Step#1 Create a new local authentication server and configure password policy; Page 21</p>

SonicWALL SMA v12.4 Assurance Activity Report

		<p>9) , Uppercases letters, and Symbols check boxes</p> <p>Password policy</p> <p>Passwords are <input type="text" value="8"/> to <input type="text" value="12"/> characters in length</p> <p>Passwords must contain at least one of the following:</p> <p><input checked="" type="checkbox"/> Lowercase letters <input checked="" type="checkbox"/> Uppercase letters</p> <p><input checked="" type="checkbox"/> Numeric digits (0-9) <input checked="" type="checkbox"/> Symbols (~!@#%&*()_+~{} v '<</p> <ul style="list-style-type: none"> • Click on Save button 	
Administrator	TLS configuration	<p>AMC:</p> <ul style="list-style-type: none"> • Login to AMC. • Navigate to "System Configuration -> SSL Settings" • Click on "Edit" link under "SSL encryption" • Modify "SSL protocols" and "SSL ciphers" • Click on "Save" button • Navigate to Monitoring -> Logging -> View Logs -> Management Audit Log 	<p>ADMIN</p> <p>Section: Configuring SSL Encryption; Page 329</p>
Administrator	FIPS mode (Enable)	<p>AMC:</p> <ul style="list-style-type: none"> • Login to AMC • Navigate to "System Configuration -> General Settings" • Click on "Edit" link under "FIPS security". • Enable FIPS mode and click on Save button. • Click on "Pending changes" link. • In the Pending changes prompt, click on the "Click here" link in the "Caution" message 	<p>CC-Addendum</p> <p>Section 3: Enabling FIPS; Page 20</p>

SonicWALL SMA v12.4 Assurance Activity Report

		<p>displayed.</p> <p>Note: The turning on FIPS will create a new self-signed certificate. Export the certificate, password protect it and save it in a safe place for future reference.</p> <ul style="list-style-type: none"> • Observe the, “Default (Workplace/access methods)” certificate used under “Certificate usage” section and export that certificate by selecting the certificate and clicking on “Export” button under “Certificates”. • Enter a password to encrypt the certificate and click on “Save” button. • Click on “Ok” and apply the pending changes. • Wait for appliance to reboot. • Login back to AMC. • Navigate to Monitoring → Logging → View Logs → Management Audit Log. 	
<p>Administrator</p>	<p>FIPS mode (Disable)</p>	<p>AMC:</p> <ul style="list-style-type: none"> • Login to AMC • Navigate to “System Configuration → General Settings” • Click on “Edit” link under “FIPS security”. • Clear the checkbox to Enable FIPS mode and click on Save button. 	<p>ADMIN Section: Disabling FIPS; Page 335</p>

SonicWALL SMA v12.4 Assurance Activity Report

		<ul style="list-style-type: none"> • Click on “Pending changes” link. • Apply Pending changes • Navigate to Monitoring -> Logging -> View Logs -> Management Audit Log. 	
<p>Administrator</p>	<p>Audit server configuration</p>	<p>AMC</p> <ul style="list-style-type: none"> • Navigate to System Configuration → Maintenance page. • Modify the URL by appending a query parameter ?advanced=1 and hit enter. • Click on “Configure...” button under Advanced/Configuration extensions. • Click on “New” button. • Add a new parameter MGMT_STRICT_CERTIFICATE_VALIDATION and set value to “true” • Click on “ok” link. • Click on “Save” button • Apply Pending Changes • Navigate to System Configuration → SSL Settings • Click on “Edit” link next to SSL Encryption • Select “TLS Versions 1.1 or 1.2” under SSL Protocols • Select the following TLS ciphers to be used under “SSL ciphers”: 	<p>CC-Addendum</p> <p>Section 10: Configure TLS settings; Page 34 Section 12: Configure external audit server (syslog); Page 36</p>

SonicWALL SMA v12.4 Assurance Activity Report

		<ul style="list-style-type: none"> ○ TLS_RSA_WITH_AES_128_CBC_SHA ○ TLS_RSA_WITH_AES_256_CBC_SHA ○ TLS_RSA_WITH_AES_128_CBC_SHA256 ○ TLS_RSA_WITH_AES_256_CBC_SHA256 <ul style="list-style-type: none"> ● Click on "save" button ● Apply pending changes 	
<p>Administrator</p>	<p>X.509 Certificate management</p> <p>Certificate Authority (CA) The entity that verifies the contents of the digital certificate and signs it indicating that the certificate is valid and correct is called the Certificate Authority (CA).</p> <p>Certificate Signing Request (CSR) An entity that wants a signed certificate or a digital certificate requests one through a CSR.</p>	<ul style="list-style-type: none"> ● Create a CA certificate or hierarchical CA (Root and Intermediate CA) to issue certificate to the TOE and Syslog server. ● Transfer the CA certificates to the workstation where AMC is accessed <p>AMC:</p> <ul style="list-style-type: none"> ● Login to AMC ● Navigate to System Configuration → SSL Settings ● Click on "Edit" link under "CA certificates" ● Click on "New" button. ● Browse the copied CA certificate in the above step and click on "Import" button. ● Navigate to Monitoring → Logging → View Logs → Management Audit Log. <p><u>Certificate Signing Request (CSR) Creation</u></p> <p>AMC:</p> <ul style="list-style-type: none"> ● Navigate to System Configuration → SSL Settings 	<p>CC-Addendum</p> <p>Section 3: Evaluated Configuration Step #8: Configure trusted Certificate Authorities (CAs); Page 31</p> <p>Section 3: Evaluated Configuration Step #9: Configure SMA web server certificate; Page 32</p>

		<ul style="list-style-type: none"> • Click "Edit" link under "SSL certificates" • Navigate to "Certificate signing requests" tab. • Click on "New" button. • Fill in valid domain name and other required fields and click on the "Save" button. • Copy the contents of the CSR text from AMC to the clipboard or into a text • Click "OK" • Navigate to Monitoring → Logging → View Logs → Management Audit Log <p><u>Certificate Signing Response (Loading CSR)</u></p> <p>BACKEND:</p> <ul style="list-style-type: none"> • Submit the CSR to a CA and download the signed TOE server certificate to the AMC workstation <p>AMC:</p> <ul style="list-style-type: none"> • Navigate to System Configuration → SSL Settings. • Click "Edit" link under "SSL certificates" • Navigate to "Certificate signing requests" tab • Click on "Process CSR response" link • Import the signed TOE certificate • Click on "Save" button • Navigate to 	
--	--	---	--

SonicWALL SMA v12.4 Assurance Activity Report

		<p>Monitoring -> Logging -> View Logs -> Management Audit Log</p> <p>BACKEND:</p> <p>Modify the TOE Server certificate and download it to the AMC workstation.</p> <p>AMC:</p> <ul style="list-style-type: none"> • Navigate to Monitoring → Logging → View Logs 	
<p>Administrator</p>	<p>Verifying and applying updates</p>	<p>Uploading a valid hotfix:</p> <ul style="list-style-type: none"> • Login to AMC • Navigate → Maintenance • Click on “Update” • Browse and upload a platform or client hotfix file by clicking on “choose file” • Click on “Install Update” • Wait for the system to reboot • Once the system is Up, login to AMC • Navigate → Monitoring → Logging → View Logs → Management Audit Log <p>Uploading an Invalid hotfix file:</p> <ul style="list-style-type: none"> • Login to AMC • Navigate → Maintenance • Click on “Update” • Browse and upload the invalid hotfix file 	<p>ADMIN</p> <p>Section: Installing System Updates; Page 325</p>

SonicWALL SMA v12.4 Assurance Activity Report

		<ul style="list-style-type: none"> Observe that upgrade fails with a message "Update failure" Navigate → Monitoring → Logging → View Logs → Management Audit Log 	
Administrator	Configuring system time	<p>AMC</p> <ul style="list-style-type: none"> Login to AMC. Navigate to System Configuration → General Settings In the Appliance options area, click Edit In the Date/time area, click Change for Current time. The Set Current Time dialog displays Enter the current date and time. Click set to apply your changes immediately. 	<p>ADMIN</p> <p>Section: To manually configure the System Time; Page 289</p>
Administrator	Configuring and modifying access banner	<p>AMC:</p> <ul style="list-style-type: none"> Login to AMC. Navigate to System Configuration → Maintenance Modify the AMC URL by appending query parameter "?advanced=1" and hit enter. Observe that, the page now displays advanced section. Click on Configure... button under Configuration extensions. Click on "New" button. Add a new CEM "ACCEPTABLE_USE_BANNER" and set value to the message you wish to display as banner during authentication. 	<p>CC-Addendum</p> <p>Section 3: Evaluated Configuration</p> <p>Step#5: Configure the Login Banner; Page 29</p>

SonicWALL SMA v12.4 Assurance Activity Report

		<p>Set Key=ACCEPTABLE_USE_BANNER and value=Welcome User! You are logging in to Management Console.</p> <ul style="list-style-type: none"> • Click on "Ok" link. • Click on "Save" button. • Apply the pending changes • Logout from AMC. • Login to AMC • Navigate to Monitoring → Logging → View Logs → Management Audit Log 	
<p>Administrator</p>	<p>Configuring termination of interactive remote session</p>	<ul style="list-style-type: none"> • Login to AMC. • Navigate to System Configuration -> Maintenance page • Modify the AMC URL by appending query parameter "?advanced=1" and hit enter • Observe that, the page now displays Advanced section • Click on "Configure..." button under "Configuration extensions" • Click on "New" button • Add a new CEM "AMC_SESSION_TIMEOUT_SECS" and set idle timeout in seconds Set Key=AMC_SESSION_TIMEOUT_SECS and value=120 • Click on "ok" link • Click on "Save" button 	<p>CC-Addendum Section 3: Evaluated Configuration Step#4: Configure idle timeout Page 27</p>

SonicWALL SMA v12.4 Assurance Activity Report

		<ul style="list-style-type: none"> • Apply the pending changes. • Logout from AMC. • Login to AMC • Stay idle for configured time. (i.e. 120 secs). • Login to AMC. • Navigate to Monitoring → Logging → View Logs → Management Audit Log 	
<p>Administrator</p>	<p>Operations related to cryptographic keys or certificates</p>	<ul style="list-style-type: none"> • Login to AMC. • Navigate to “System Configuration → SSL Settings” • Click on “Edit” link under “SSL Certificates” • Click on “New” → “Create self-signed certificate...” • Fill in the required inputs and click on “Save” button • Click on certificate link under “Certificate usage” for AMC • Select newly created self-signed certificate and click on “ok” link • Navigate to Monitoring → Logging → View Logs → Management Audit Log • Navigate to “System Configuration → SSL Settings” • Click on “Edit” link under “SSL Certificates” 	<p>CC-Addendum Section 3: Evaluated Configuration Step# 9: Configure the SMA web server certificate; Page 32</p>

SonicWALL SMA v12.4 Assurance Activity Report

		<ul style="list-style-type: none"> • Click on certificate link under “Certificate usage” for AMC • Select any other self-signed certificate and click on “ok” link • Now, select above created self-signed certificate under “Certificates” and click on “Delete” button. • Navigate to Monitoring -> Logging -> View Logs -> Management Audit Log 	
<p>Administrator</p>	<p>Administrative login</p>	<p><u>Successful administrative login:</u></p> <p>AMC:</p> <ul style="list-style-type: none"> • Login to AMC as a cadmin with valid password. • Navigate to Monitoring -> Logging -> View Logs -> Management Audit Log. <p>CLI:</p> <ul style="list-style-type: none"> • Login to appliance console • Login as cadmin with valid credentials. <p><u>Unsuccessful administrative login:</u></p> <p>AMC:</p> <ul style="list-style-type: none"> • Login to AMC as a cadmin with invalid password • Now login to AMC as a cadmin with valid password 	<p>CC-Addendum</p> <p>Section: 2.3 Accessing Secure Mobile Access Management Console; Page 18</p>

SonicWALL SMA v12.4 Assurance Activity Report

		<ul style="list-style-type: none"> Navigate to Monitoring -> Logging -> View Logs -> Management Audit Log. 	
Administrator	Account management	<p><u>Creation of a new user:</u></p> <p>AMC:</p> <ul style="list-style-type: none"> Login to AMC. Navigate to "Security Administration -> Users and Groups -> Local Accounts" Tab. Click on "New -> User". Create a local user by entering name as "user1" and enter "password". Navigate to "System Configuration -> General Settings." Navigate to "Administrators -> Edit" tab. Click on "New -> Administrator". Create a new admin by selecting "user1-local user" and any "Role" Click save Apply Pending Changes Navigate to Monitoring -> Logging -> View Logs -> Management Audit Log. <p><u>Disabling of user account by administrative action:</u></p> <ul style="list-style-type: none"> Navigate to "Security Administration -> Users and Groups 	<p>CC-Addendum</p> <p>Section 3: Evaluated Configuration</p> <p>Step#2: Create a new local administrator; Page 23</p>

		<p>→ Local Accounts” tab</p> <ul style="list-style-type: none"> • Click on above created admin user “user1”. • Disable the option "User is enabled". • Click on “Save” button • Apply Pending Changes • Navigate to Monitoring → Logging → View Logs → Management Audit Logs. <p><u>Deletion of existing account:</u></p> <ul style="list-style-type: none"> • Login to AMC. • Navigate to "System Configuration -> General Settings -> Administrators" tab. • Select the user • Click → Delete • Click Save and Apply Pending Changes • Navigate to Monitoring → Logging → View Logs → Management Audit Log <p><u>Reset of User Password:</u></p> <ul style="list-style-type: none"> • Login to AMC. • Navigate to “System Configuration → General Settings” • Click on “Edit” link under Administrators. • Click on Primary Admin. • Enter current admin password under “Verify administrator password” textbox. 	
--	--	--	--

SonicWALL SMA v12.4 Assurance Activity Report

		<ul style="list-style-type: none"> • Enable "Reset password for this administrator". • Enter new password and confirm password by re-entering it. • Click on "Save" button. • Navigate to Monitoring → Logging → View Logs → Management Audit Log. 	
<p>Administrator</p>	<p>Failure to establish a TLS session</p>	<p><u>Failure to establish a TLS Session</u></p> <ul style="list-style-type: none"> • Login to AMC in Browser1. • Navigate to "System Configuration → SSL Settings". • Click on "Edit" link under "SSL Encryption". • Configure "TLS version 1.2 only" under "SSL protocols" and select ciphers to be used under "SSL ciphers". • Apply pending changes. <p><u>Browser2:</u></p> <ul style="list-style-type: none"> • Open Browser2 and disable TLS version 1.2 • Try to Login to AMC from Browser2. <p><u>Browser1:</u></p> <ul style="list-style-type: none"> • Navigate to Monitoring -> Logging -> View Logs -> Management Message Logs. 	<p>CC-Addendum</p> <p>Section 3: Evaluated Configuration</p> <p>Step#10: Configure TLS settings; Page 34</p>

SonicWALL SMA v12.4 Assurance Activity Report

Administrator	Unsuccessful attempt to validate an X509 certificate	<u>Syslog Server:</u> <ul style="list-style-type: none">• Create a syslog certificate with CN not matching the hostname or IP Address of the syslog and no SAN extension is added to the certificate• Update the syslog server configuration with the new certificate• Restart the syslog server <u>AMC:</u> <ul style="list-style-type: none">• Login to AMC using admin credentials• Verify that connection fails between appliance and syslog server and the audit logs in Monitoring → Logging → Management Audit Logs	CC-Addendum Section 3: Evaluated Configuration Step #12: Configure external audit server; Page 36
----------------------	---	--	---

2.1.1.3 Testing Assurance Activities

Testing Assurance Activities:

- (1) The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the table of audit events and administrative actions listed above. This should include all instances of an event: for instance, if there are several different I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism.
- (2) 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.
- (3) 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.

Testing Implementation Details/Results:

- (1) The evaluator collected all the auditable events evidence in the [AUDITR] and confirmed that the TOE was able to correctly generate audit records for the events listed in the Table 5.
- (2) The evaluator used custom Tool to test the TLS communication between the TOE and TLS Tool. The custom TLS Tool was used to test the supported cryptographic protocols of the TOE as mentioned in the ST by establishing and/or terminating the TLS session between the TOE and the custom TLS Tool.
- (3) As part of the overall testing effort, the evaluator verified that each secure channel generates appropriate audit records and that each audit record matches the format specified in the guidance.

2.1.2 FAU_GEN.2 Audit Data Generation

2.1.2.1 TSS Assurance Activities

TSS Assurance Activities:

- (1) 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.
- (2) The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access.
- (3) The evaluator shall examine the TSS to ensure it describes whether the TOE is a standalone TOE that stores audit data locally or a distributed TOE that stores audit data locally on each TOE component or a distributed TOE that contains TOE components that cannot store audit data locally on themselves but need to transfer audit data to other TOE components that can store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs it contains a list of TOE components that store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs that contain components which do not store audit data locally but transmit their generated audit data to other components it contains a mapping between the transmitting and storing TOE components.
- (4) The evaluator shall examine the TSS to ensure that it details the behaviour of the TOE when the storage space for audit data is full. When the option 'overwrite previous audit record' is selected this description should include an outline of the rule for overwriting audit data. If 'other actions' are chosen such as sending the new audit data to an external IT entity, then the related behaviour of the TOE shall also be detailed in the TSS.

The evaluator shall examine the TSS to ensure that it details whether the transmission of audit information to an external IT entity can be done in real-time or periodically. In case the TOE does not perform transmission in real-time the evaluator needs to verify that the TSS provides details about what

Assurance Activity Report

event stimulates the transmission to be made as well as the possible as well as acceptable frequency for the transfer of audit data.

TSS Implementation Details/Results: N/A

2.1.2.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Implementation Details/Results: N/A

2.1.2.3 Testing Assurance Activities

Testing Assurance Activities:

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

Testing Implementation Details/Results: N/A

2.1.3 FAU_STG_EXT.1 Protected Audit Event Storage

2.1.3.1 TSS Assurance Activities

TSS Assurance Activities:

- (1) 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.
- (2) The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access.
- (3) 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.
- (4) The evaluator shall examine the TSS to ensure that it details the behaviour of the TOE when the storage space for audit data is full. When the option 'overwrite previous audit record' is selected, this description should include an outline of the rule for overwriting audit data. If 'other actions' are chosen such as sending the new audit data to an external IT entity, then the related behaviour of the TOE shall also be detailed in the TSS.
- (5) The evaluator shall examine the TSS to ensure that it details whether the transmission of audit information to an external IT entity can be done in real-time or periodically. In case the TOE does not perform transmission in real-time the evaluator needs to verify that the TSS provides details about what event stimulates the transmission to be made as well as the possible as well as acceptable frequency for the transfer of audit data.

TSS Implementation Details/Results:

- (1) The evaluator noted that the trusted channel with an audit server is secured with TLS. The evaluator examined the ST Section 7.1 and found the following description: "The TOE is designed to securely forward audit records to a designated external audit server over a persistent trusted channel. This external audit server is authenticated by checking X.509v3 certificate and secured with a TLS protocol."
- (2) The ST Section 7.1 mentions that all audit events are recorded locally on the TOE and can also be securely transferred to an external audit server. The audit records are stored locally on a separate partition (/var/log) of size 128GB. The TOE creates different log file for different activities (Management, Web Proxy, SSL etc) and each log exists as a set of 168 log files that collectively operate as a circular archive. In case the local partition (/var/log) gets full, all archived log files older than 7 days are deleted. The evaluator noted from the

Assurance Activity Report

Section 7.1 of the ST, that the viewing and clearing of the audit logs are strictly protected by role-based access, where access is restricted to Security Administrators with appropriate permission. This confirms that the logs are protected against unauthorized access for modifications or deletion.

- (3) The evaluator noted from the Section 7.1 that the TOE is a standalone network device and is consistently described as such throughout the ST. The evaluator examined the ST Section 7.1 and noted that it includes description that the TOE stores the audit log locally and also can be configured to send the log files to an external syslog server.
- (4) The ST Section 7.1 describes that by default, local audit trail is limited to 7 days of audit records and available disk space in /var/log. On-device audit records exist in as a set of files that operate as a circular log file; the state of audit repository is periodically checked, when it is detected that it is getting full, logs older than 7 days are deleted. This matches with the SFR selection of "delete all log files older than 7 days".
- (5) The ST Section 7.1 describes that TOE can be configured to transfer the audit records to the remote syslog server in real time. The ST Section 7.1 states "When configured, the TOE uploads audit records in syslog (RFC 5424) format as they are generated without any delay."

2.1.3.2 Guidance Assurance Activities

Guidance Assurance Activities:

- (1) 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.
- (2) The evaluator shall also examine the guidance documentation to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.
- (3) The evaluator shall also ensure that the guidance documentation describes all possible configuration options for FAU_STG_EXT.1.3 and the resulting behaviour of the TOE for each possible configuration. The description of possible configuration options and resulting behaviour shall correspond to those described in the TSS.

Guidance Implementation Details/Results:

- (1) The evaluator examined the CC-Addendum, Section 3 Evaluated configuration: Step#10. Configure TLS settings Page 24, and Step#12. Configure external audit server (syslog) Page 36 and ensured that it describes how to establish a trusted channel to the external audit server using TLS protocol. The CC-Addendum provides the necessary steps to integrate TOE and Syslog Server to establish a secure communication between them.
- (2) The ST Section 7.1 describes that the TOE can be configured to transfer the audit records to the remote syslog server without any delay. The ST Section 7.1 states "When configured, the TOE uploads audit records in syslog (RFC 5424) format as they are generated without any delay."
- (3) The evaluator examined the ADMIN, Appendix E Log File Output Formats; Sub-section: Log Rotation Procedure (Page 588) and ensured that it describes configuration option for FAU_STG_EXT.1.3 `[[delete all log files older than 7 days]]` and ensured that the TOE overwrites the oldest log in the file when it meets the conditions specified.

2.1.3.3 Testing Assurance Activities

Testing Assurance Activities:

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

Test 1:

Assurance Activity Report

- 1) The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided.
- 2) 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.
- 3) The evaluator shall record the particular software (name, version) used on the audit server during testing.
- 4) The evaluator shall verify that the TOE is capable of transferring audit data to an external audit server automatically without administrator intervention.

Test 2:

The evaluator shall perform operations that generate audit data and verify that this data is stored locally.

The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.3.

Depending on the configuration this means that the evaluator has to check the content of the audit data when the audit data is just filled to the maximum and then verifies that

- 1) The audit data remains unchanged with every new auditable event that should be tracked but that the audit data is recorded again after the local storage for audit data is cleared (for the option 'drop new audit data' in FAU_STG_EXT.1.3).
- 2) The existing audit data is overwritten with every new auditable event that should be tracked according to the specified rule (for the option 'overwrite previous audit records' in FAU_STG_EXT.1.3)
- 3) The TOE behaves as specified (for the option 'other action' in FAU_STG_EXT.1.3).

Test 3:

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

Test 4:

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

Testing Implementation Details/Results:

The evaluator followed the procedures outlines in the guidance to configure TOE audit functionality.

Test 1:

- (1) The evaluator followed the CC-Addendum, Section 3 Step #12 Configuring external audit server to establish a secure connection between the TOE and the audit server.
- (2) The evaluator used Wireshark software and a mirroring port on a core switch to view and capture relevant network traffic exchange between the TOE and the remote audit server. The evaluator examined network traffic, observed a successful TLS handshake and encrypted data sent between the TOE and the audit server, and concluded that a secure channel was successfully established between the TOE and the remote audit server. During this period, the evaluator observed audit records updated on the remote audit server and concluded that these audit records were sent as part of encrypted traffic. See test case PP-8A for audit details.
- (3) The evaluator used syslog-ng-3.29 as the external audit server for the testing.

SonicWALL SMA V12.4

Assurance Activity Report

- (4) Upon enabling the secure logging configuration between the TOE and the Audit Server, the evaluator noted audit events messages showing up on the audit server event's log file for every action executed on the AMC and CLI Commands issued, without needing any administrator intervention.

Test 2:

- (1) The evaluator executed the instructions from the CC-Addendum and confirmed that logs are created locally on the TOE for all the instructions that were executed as given in the CC-Addendum. The evaluator confirmed this by checking the logs from **Console → /var/log/syslog** as well as through **AMC → Monitoring → Logging → Management Audit Logs**. The evaluator used the **Section 4 - Auditable Events** of the CC-Addendum to compare the format and details of the audit logs generated by the TOE.
- (2) The evaluator ran a custom script to fill the disk partition where the log files are stored (/var/log). As mentioned in the ST Section 7.2 and Section 6.1.1 FAU_STG_EXT.1.3, the evaluator observed that all log files older than 7 days were deleted to free up the disk space.

Test 3 and Test 4: not applicable to the TOE

2.2 Cryptographic Support (FCS)

2.2.1 FCS_CKM.1 Cryptographic Key Generation (for asymmetric keys)

2.2.1.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The ST Section 7.2 Table-14 entry for FCS_CKM.1 states:

“Generating 2048-bit and 3072-bit RSA key pairs validated conforming to FIPS186-4.”

Based on the above information, the evaluator confirmed that the TSS identifies the key size supported by the TOE for the Cryptographic Key Generation.

2.2.1.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Implementation Details/Results:

The TOE implements TLS client and TLS server that use RSA and EC key authentication for the trusted channel. The evaluator examined the CC-Addendum, **Section 3 Evaluated configuration Step#10 Configure TLS settings; Page 34** and verified that it contains instructions for the administrator to configure the supported cipher suites for TOE for the TLS communication.

2.2.1.3 Testing Assurance Activities

Testing Assurance Activities:

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

1. Key Generation for FIPS PUB 186-4 RSA Schemes

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

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

a) Random Primes:

- Provable primes
- Probable primes

b) Primes with Conditions:

- Primes p_1, p_2, q_1, q_2, p and q shall all be provable primes
- Primes p_1, p_2, q_1, q_2, p and q shall be provable primes and p and q shall be probable primes

Assurance Activity Report

- Primes p_1, p_2, q_1, q_2, p and q shall all be probable primes

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

2. Key Generation for Elliptic Curve Cryptography (ECC)

FIPS 186-4 ECC Key Generation Test

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

FIPS 186-4 Public Key Verification (PKV) Test

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

3. 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 :

- $\text{len}(q)$ bit output of RBG where $1 \leq x \leq q-1$
- $\text{len}(q) + 64$ bit output of RBG, followed by a mod $q-1$ operation and a $+1$ operation, where $1 \leq x \leq q-1$.

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

- $g \neq 0, 1$

SonicWALL SMA V12.4

Assurance Activity Report

- q divides $p-1$
- $g^q \bmod p = 1$
- $g^x \bmod p = y$

for each FFC parameter set and key pair.

4. FFC Schemes using “safe-prime” groups.

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

Note: TD0580 https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0580 was applied.

Testing Implementation Details/Results:

The TOE utilizes cryptographic primitives validated according to NIST Cryptographic Algorithm Validation Program, ACVP certificates A1338, A1352, and A1358. These algorithm certificate identify SMA1000 operating system and CPUs included in SonicWALL SMA v12.4 SMA 6210, SMA 7210, and SMA 8200v appliances. These algorithm certificates covers RSA, EC, AES, SHA, HMAC, DRBG and TLS key establishment functionality and includes avcrypto (kernel) A1338, libcrypto (OpenSSL) A1352, and OpenJDK (Java) A1358 implementations.

- (1) The TOE performs the cryptographic RSA-Based key generation via the cryptographic module, which was tested according to the Cryptographic Algorithm Validation Program and was awarded the certificate number #A1352 covering libcrypto (OpenSSL), and #A1358 covering the OpenJDK (Java) implementations.
- (2) The TOE performs the cryptographic ECC-Based key generation and key verification via the cryptographic module, which was tested according to the Cryptographic Algorithm Validation Program and was awarded the certificate number #A1352 covering libcrypto (OpenSSL), and #A1358 covering the OpenJDK (Java) implementations
- (3) The TOE doesn't support FFC for FCS_CKM.1. This activity is not applicable.
- (4) The TOE doesn't support Diffie-Hellman group 14 and/or safe-prime groups for FCS_CKM.1. This activity is not applicable.

Assurance Activity Report

2.2.2 FCS_CKM.2 Cryptographic Key Establishment

2.2.2.1 TSS Assurance Activities

TSS Assurance Activities:

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

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

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

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

Note: TD0580 https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0580 was applied.

TSS Implementation Details/Results:

The evaluator noted from the ST section 6.1.2, FCS_CKM.2, and confirmed that the key establishment key scheme provided in the FCS_CKM.2 corresponds to the key generation scheme identified in FCS_CKM.1.

The ST, Section 6.1.2 (FCS_CKM.2) states that the TOE performs cryptographic key establishment in accordance with a specific cryptographic key establishment method:

- 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";]

The evaluator checked the ST Section 7.2 Table 15 and confirmed that the FCS_CKM.2 identifies the Elliptic curve-based key establishment scheme only and no other scheme is specified in the table as supported by TOE for FCS_CKM.2. The following services and the protocols are used with RSA key establishment.

Scheme	SFR	Service	Protocol
ECC	FCS_TLSS_EXT.1	Administration (AMC)	TLS
	FCS_TLSC_EXT.1	Remote Audit (Syslog)	
	FCS_TLSS_EXT.2	VPN (VPN connect tunnel)	

2.2.2.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Assurance Activities Details/Results:

Assurance Activity Report

The evaluator examined the CC-Addendum, **Section 3 Evaluated configuration Step#10 Configure TLS settings; Page 34** and verified that it contains procedure for the administrator to configure the TOE to use the selected key establishment scheme(s).

2.2.2.3 Testing Assurance Activities

**Testing Assurance Activities:
Key Establishment Schemes**

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

SP800-56A Key Establishment Schemes

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

Function Test

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

The evaluator shall obtain the DKM, the corresponding TOE's public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE id fields. If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret.

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

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

Validity Test

The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) (1) 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.

Assurance Activity Report

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

The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOE's results with the results using a known good implementation verifying that the TOE detects these errors.

RSA-based key establishment

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

FFC Schemes using 'safe-prime' groups

The evaluator shall verify the correctness of the TSF's implementation of safe-prime groups by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses safe-prime groups. This test must be performed for each safe-prime group that each protocol uses.

Note: TD0580 https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0580 was applied.

Testing Assurance Activities Details/Results:

Evaluator checked the ST and found that it is listing the relevant SFRs: FTP_TRP.1/Admin and FTP_ITC.1 with TLS as the only selection for the trusted channel and trusted path. These selections correspond to FCS_TLSC_EXT.1, FCS_TLSS_EXT.2 and FCS_TLSS_EXT.1 SFRs. The TOE performs ECC-Based key Elliptic curve-based key establishment schemes that meet the NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"; generation and key verification via the cryptographic module, which was tested according to the Cryptographic Algorithm Validation Program and was awarded the certificate number #A1352 covering Libcrypto (OpenSSL), and #A1358 covering the OpenJDK (Java) implementations.

2.2.3 FCS_CKM.4 Cryptographic Key Destruction

2.2.3.1 TSS Assurance Activities

TSS Assurance Activities:

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

SonicWALL SMA V12.4

Assurance Activity Report

- (3) The evaluator shall check to ensure the TSS identifies how the TOE destroys keys stored as plaintext in non-volatile memory, and that the description includes identification and description of the interfaces that the TOE uses to destroy keys (e.g., file system APIs, key store APIs).
 Note that where selections involve 'destruction of reference' (for volatile memory) or 'invocation of an interface' (for non-volatile memory) then the relevant interface definition is examined by the evaluator to ensure that the interface supports the selection(s) and description in the TSS. In the case of non-volatile memory the evaluator includes in their examination the relevant interface description for each media type on which plaintext keys are stored. The presence of OS-level and storage device-level swap and cache files is not examined in the current version of the Evaluation Activity.
- (4) 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.
- (5) 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.
- (6) Where the ST specifies the use of "a value that does not contain any CSP" to overwrite keys, the evaluator examines the TSS to ensure that it describes how that pattern is obtained and used, and that this justifies the claim that the pattern does not contain any CSPs.

TSS Implementation Details/Results:

The evaluator examined the ST Section 7.2 TSS and noted that 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.

- (1) The evaluator confirmed that Table 16 in the Section 7.2 contains the list of plaintext CSPs, name of the keys, key generation process, the purpose of the keys, and storage location of each key, along with the method of zeroization and situation of zeroization.

Table 7: SonicWALL SMA v12.4 CSPs

Identifier	Name	Generation / Algorithm	Purpose	Storage Location	Zeroization Summary
TLS-AMC-Priv	Private Key	PKCS1v1_5 / RSA or FIPS PUB 186-4 Appendix B.4/ECC	X509 private key used for certificate-based authentication	RAM (plain text) Disk (ciphertext)	Single direct overwrite consisting of zeros followed by a read-verify action.
TLS-SENC	TLS Session Keys	Generated using TLS KDF	Symmetric keys for TLS	RAM (plain text)	Cleared when device is powered down or as part of session termination. Overwritten by a new value.

SonicWALL SMA V12.4

Assurance Activity Report

AUTH-PW	Authentication Passwords	SHA256	Credentials used to authenticate the administrator login.	Disk (cipher text)	Hashed passwords exist in a local database and replaced when changed and saved. The passwords are stored in the ciphertext (hash and salt) form only. Overwritten by a new value
				RAM (cipher and plain text)	Passwords in RAM are zeroized when creating / resetting the password. Both clear text and encrypted forms are stored in RAM. Overwritten by new value.
DRBG-EI	PRNG Seed key	/dev/random	Seed key for PRNG	RAM (plain text)	Cleared when device is powered down or during reboot by the new seed.
OS-KEK	Keystore encryption key	Platform	Used to encrypt CSPs in certificate storage	RAM (plain text)	In RAM, cleared when device is powered down or during reboot.
				Disk (plain text)	On disk, overwritten by zeroization.

- (2) The evaluator confirmed that Table 16 in the Section 7.2 contains the list of all keys generated and used for the TOE-specific secure channels and protocols. The description of keys and storage locations provides a general understanding of the implemented functionality. However, the evaluators have no direct access to the underlying implementation and have no way to verify these claims other than to confirm that the TSS section includes such claims and that they are plausible.
- (3) The evaluator checked the ST Section 7.2 and confirmed that the TSS section contains information on the key destruction requirements as the following: "The TOE is designed to destroy Critical Security Parameters (CSPs) when no longer required for use to mitigate the possibility of disclosure. At various times during TOE operation (e.g., an active TLS session) CSPs are present in RAM in plain text, then de-allocated and cleared from memory when no longer needed (e.g., on TLS session termination). Some CSPs (e.g., long term private keys) are also stored on disk and cleared when no longer used." **Please refer to Table 7 above for the details on the identification and description of the interfaces that the TOE uses to destroy keys.**
- (4) The ST Table 16 lists all keys that are stored in plain-text and explains how they are cleared. TLS-AMC-Priv is listed as stored as ciphertext. The TSS explains that it is a private key associated with TOE's X.509 certificate and that it is "encrypted java keystore that is in turn protected with OS-KEK".

SonicWALL SMA V12.4

Assurance Activity Report

(5) The evaluator checked the TSS and noted that it does not identify any specific configuration or circumstances that would prevent key destruction from functioning correctly. The evaluator determined this to be acceptable as the TOE software and hardware architecture does not include known problematic technologies (e.g., wear leveling).

(6) The evaluator confirmed that this selection is not made within the ST and is not applicable.

2.2.3.2 Guidance Assurance Activities

Guidance Assurance Activities:

A TOE may be subject to situations that could prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS (and any other supporting information used). The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer.

For example, when the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-leveling 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 and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

Guidance Assurance Activities Details/Results:

As per the ST, The TOE is not subjected to any situations that could prevent or delay key destruction.

The evaluator checked the ADMIN Section 5 Network and Authentication Configuration; Sub-section: Managing CA Certificates Page 162; and Section 7 System Administration; Sub-section: Disabling FIPS Page 335 and confirmed that the guidance document contains instructions to permanently delete the unused or unwanted keys and certificates from the TOE.

2.2.3.3 Testing Assurance Activities

Testing Assurance Activities: None

Testing Assurance Activities Details/Results: N/A

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

2.2.4.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The evaluator checked the ST Section 7.2, Table 15 and confirmed that the TOE uses the cryptographic algorithm AES for data encryption and decryption used in CBC, GCM mode with 128-bit, 256-bit key sizes validated conforming to FIPS PUB 197.

2.2.4.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Assurance Activities Details/Results:

Assurance Activity Report

The evaluator checked the ADMIN Section 7 System Administration; Sub-section: SSL Encryption Page 328; and checked the CC-addendum section Configure SMA web server certificate Page 21 and had determined that these sections instruction the administrator on how to configure the TOE to use the desired security levels and their supported ciphers and TLS protocol versions.

2.2.4.3 Testing Assurance Activities

Testing Assurance Activities:

Specific algorithm tests are detailed in the Supporting Document Section 2.2.4.1

Testing Assurance Activities Details/Results:

The ST Section 7.2, Table 15 claims that the TOE uses the cryptographic algorithm AES for data encryption and decryption used in CBC, GCM mode with 128-bit, 256-bit key sizes. The evaluator confirmed that the TOE's AES encryption/decryption implementation was rewarded the CAVP certificates: A1338, A1358, A1352 using the exact version of the cryptographic module following appropriate installation and usage guidance.

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

2.2.5.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The evaluator checked the ST Section 7.2, Table 15 and confirmed that the TOE uses RSA signature generation and verification according to RSASSA-PKCS1v1_5 with 2048-bit and 3072-bit key sizes and utilizing SHA2-256 and SHA2-384.

In addition, the table 15 in ST section 7.2 specifies that the TOE uses the ECDSA signature generation and verification using P-256, P-384 curves with the SHA-256, SHA-384 hash algorithms.

2.2.5.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Assurance Activities Details/Results:

The evaluator checked the ADMIN Section 7 System Administration; Sub-section: SSL Encryption Page 328; and checked the CC-addendum section Configure SMA web server certificate Page 21 and had determined that these sections instruction the administrator on how to configure the TOE to use a predefined cryptographic security level. The ADMIN Section System administration mentions that all security levels use only US government-recommended (FIPS 140-2 Compliant) cryptographic algorithms, and that the TOE's administrator has not ability to configured specific cryptographic algorithms suites.

2.2.5.3 Testing Assurance Activities

Testing Assurance Activities:

Assurance Activity Report

Specific algorithm tests are detailed in the Supporting Document Section 2.2.5.1

Testing Assurance Activities Details/Results:

The evaluator verified that the TOE's Signature Generation and Verification implementation was rewarded the CAVP certificates: #A1358 and #A1352.

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

2.2.6.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The evaluator checked the ST Section 7.2 and confirmed that the TSS identifies hashing functionality used by other cryptographic functionality, specifically:

- Table 15 entry for FCS_COP.1/Hash details Hashing using SHA-1, SHA-256 and SHA-384 validated conforming to FIPS 180-4, Secure Hash Standard (SHS).
- Table 15 entry for FCS_COP.1/KeyedHash lists
 - Keyed hash HMAC-SHA1, HMAC-SHA256, validated conforming to FIPS 198, Keyed-Hash Message Authentication Code (HMAC).
 - Supported cryptographic key sizes: 160 and 256 bits and message digest sizes: 160, 256 bits.
 - Keyed hash use matches validated hash algorithms implemented by the module. The evaluator examined these descriptions and concluded that they are consistent and match functionality claimed in CAVP certificates listed below.

The evaluator examined these descriptions and concluded that they are consistent and match functionality claimed in the CAVP certificates: A1338, A1358 and A1352

2.2.6.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Assurance Activities Details/Results:

The evaluator examined the CC-Addendum, **Section 3 Evaluated configuration Step# 10: Configure TLS settings; Page 34** and confirmed that configuration required to configure the required hash sizes for the generation of the TOE's cryptographic key pair is present in the guidance document.

The evaluator checked the ADMIN Section 7 System Administration; Sub-section: SSL Encryption Page 328; and had determined that these sections instruction the administrator on how to configure the TOE to use a predefined cryptographic security level when communicating with third-part IP hosts. The ADMIN Section System administration mentions that all security levels use only US government-recommended (FIPS 140-2 Compliant) cryptographic algorithms, and that the TOE's administrator has not ability to configured specific cryptographic algorithms suites.

2.2.6.3 Testing Assurance Activities

Testing Assurance Activities:

Specific algorithm tests are detailed in the Supporting Document Section 2.2.6.3.

Testing Assurance Activities Details/Results:

Assurance Activity Report

The evaluator verified that the TOE's hashing implementation as validated by the CAVP certificates (A1338, A1358, A1352) using the exact version of the cryptographic module following appropriate installation and usage guidance. This validates the claims of conformance for the TOE's hashing functionality to ISO/IEC 10118-3:2004 "Secure Hash Standard" within its operational environment.

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

2.2.7.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The evaluator examined the ST Section 7.2 Table 15 and noted that it references the following CAVP Certificates (A1338, A1358, A1352) that specifies the algorithm as HMAC-SHA1 and HMAC-SHA2-256, supported cryptographic key sizes as 160 bits, 256 bits and message digest sizes as 160 and 256 bits. It matches with the ST Section 6.1.2.

2.2.7.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Assurance Activities Details/Results:

The evaluator checked the ADMIN Section 7 System Administration; Sub-section: SSL Encryption Page 328; and had determined that the section instructs the administrator on how to configure the TOE to use a predefined cryptographic security level. The ADMIN Section System administration mentions that all security levels use only US government-recommended (FIPS 140-2 Compliant) cryptographic algorithms, and that the TOE's administrator has not ability to configured specific cryptographic algorithms suites.

2.2.7.3 Testing Assurance Activities

Testing Assurance Activities:

Specific algorithm tests are detailed in the Supporting Document Section 2.2.7.2.

Testing Assurance Activities Details/Results:

The evaluator verified that the TOE's hashing implementation as validated by CAVP certificates (A1338, A1358, A1352) using the exact version of the cryptographic module following appropriate installation and usage guidance. This validates the claims of conformance for the TOE's keyed hashing functionality to ISO/IEC 9797-2:2011 "HMAC Standard" within its operational environment.

Assurance Activity Report

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

2.2.8.1 TSS Assurance Activities

TSS Assurance Activities:

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.

TSS Assurance Activities Details/Results:

The evaluator examined the ST Section 7.2 and determined the DRBG type and the entropy sources from the following statement in the ST that “The TOE uses a software-based random bit generator (CTR_DRBG as implemented by avcrypto) that complies with NIST SP 800-90A for all cryptographic operations. Each DRBG instance is seeded with full entropy sourced from CPU Time Jitter Based Non-Physical TRNG (aka Jitter Entropy) with a minimum of 256-bits of entropy. The Jitter Entropy is compiled into a Linux Kernel Module that is loaded at boot time as part of DRBG initialization. This entropy source is used solely for the purpose of seeding and reseeding PRNGs (i.e., CTR_DRBG in FIPS mode). Jitter Entropy is a software-based mechanism that relies on the timing of unpredictable events. These events take the form of CPU execution jitter – measurable differences in the time it takes the CPU to execute a given set of instructions. Jitter Entropy is on-demand entropy source and does not accumulate entropy or preserve entropy across system reboots.” The detailed entropy justification is provided in a separate [ENT] document.

2.2.8.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Assurance Activities Details/Results:

No configuration is required for evaluated implementation of the RNG functionality. Based on the findings that RNG is not configurable on the SMA appliance, this activity is considered satisfied.

2.2.8.3 Testing Assurance Activities

Testing Assurance Activities:

Specific algorithm tests are detailed in the Supporting Document Section 2.2.8.3.

Testing Assurance Activities Details/Results:

The evaluator verified that the TOE uses a software-based entropy source to seed a CTR_DRBG (AES-256) random bit generator as validated by the CAVP certificate #A1338 that meets NIST SP 800-90A “Recommendation for Random Number Generation Using Deterministic Random Bit Generators”.

2.2.9 FCS_TLSC_EXT.1 TLS Client Protocol

2.2.9.1 FCS_TLSC_EXT.1.1

2.2.9.1.1 TSS Assurance Activities

TSS Assurance Activities:

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.

Assurance Activity Report

TSS Implementation Details/Results:

The ST, Section 7.2 describes the implementation of the TLS and lists the following ciphersuites:

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

as the supported cipher suites.

The evaluator verified that the cipher suites specified match those listed for this component in Section 6.1.2.

2.2.9.1.2 Guidance Assurance Activities

Guidance Assurance Activities:

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.

Guidance Implementation Details/Results:

The evaluator examined the CC-Addendum, **Section 3 Evaluated configuration Step# 10: Configure TLS settings; Page 34** and confirmed that the guidance document contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.

2.2.9.1.3 Testing Assurance Activities

Testing Assurance Activities:

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

Test 1: The evaluator shall establish a TLS connection using each of the cipher suites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an 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 2: The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage field and verify that a connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field, and a connection is not established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.

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

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

- a) The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection.
- b) Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.
- c) [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension the evaluator shall configure the server to perform an ECDHE or DHE key exchange in the TLS connection

Assurance Activity Report

using a non-supported curve/group (for example P-192) and shall verify that the TOE disconnects after receiving the server's Key Exchange handshake message.

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

- a) Change the TLS version selected by the server in the Server Hello to a non-supported TLS version and verify that the client rejects the connection.
- b) [conditional]: If using DHE or ECDH, modify the signature block in the Server's Key Exchange handshake message, and verify that the handshake does not finished successfully, and no application data flows. This test does not apply to cipher suites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.

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

- a) Modify a byte in the Server Finished handshake message and verify that the handshake does not finish successfully, and no application data flows.
- b) Send a garbled message from the server after the server has issued the ChangeCipherSpec message and verify that the handshake does not finish successfully, and no application data flows.

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.

Testing Implementation Details/Results:

Test 1: The TOE implements trusted channel with external audit server. In this configuration, the TOE acts as a TLS client and a custom TLS Tool is used in the place of external audit server to test the different cipher suites. The following TLS ciphers are supported in the evaluated configuration:

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

In test case PP-20A, the evaluator established a TLS connection using each of the cipher suites specified by the above requirement as given in the ST. A custom test tool was used as the TLS Server, configured to support only the above listed ciphersuites for the TLS negotiation. The evaluator configured the IP Address and the port number of the custom TLS tool as one of the external audit servers in the Logging section of the TOE via AMC. The TOE then initiated connection to the custom TLS Tool server and successfully established connection with the TLS Tool for each supported cipher suite. Packet capture was then used to observe the handshake and to confirm that the selected ciphersuite was successfully negotiated between the TOE and the custom tool.

Test 2: As part of test case PP-20B, the custom test tool was used as the TLS Server, configured with proper X.509 certificate with valid server certificate with Server Auth OID in the extendedKeyUsage extension in the certificate and selected to only support the above listed ciphersuites for the TLS negotiation. Packet capture was then used to observe the handshake and to confirm that the selected ciphersuite was successfully negotiated between the TOE and the custom tool. The same test case was executed with otherwise valid certificate without the extendedKeyUsage extension and invalid extendedKeyUsage extension (Client Auth OID) and confirmed the failure of the TLS handshake between the TOE and the Custom TLS Tool. Packet Capture was then used to observe the handshake failure to confirm that the certificate was not for intended purpose.

Test 3: As part of test case PP-20C, the custom test tool was configured with wrong server certificate (ECDSA Server Certificate) for the TLS negotiation. The evaluator observed the disconnection from the TOE after receiving the Server Certificate handshake message.

Test 4a: As part of test case PP-20D, the custom test tool was configured with TLS_NULL_WITH_NULL_NULL for the TLS negotiation. The evaluator observed the connection denied message from the TOE.

Test 4b: As part of test case PP-14G, configured the TLS Tool to modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake

SonicWALL SMA V12.4

Assurance Activity Report

message. The evaluator confirmed from the packet capture that the TOE rejected the connection after receiving the Server Hello.

Test 4c: As part of test case PP-20H: the evaluator setup the TOE to send audit events to a customer TLS tool which was set to presents the non-supported Elliptic Curves P-192, the evaluator observed that the TOE disconnects after receiving the server's Key Exchange handshake message.

Test 5: The evaluator performed the following modifications to the traffic and observed the TOE rejection of the TLS handshake for every configuration listed below:

- a) As part of test case PP-20E, configured the TLS Tool with "TLS version to non-supported" value and confirmed that the TOE rejected the connection.
- b) As part of test case PP-20F, configured TLS Tool to modify the signature block in the Server's Key Exchange handshake message, and verified that the TOE rejected the Server Key Exchange handshake message

Test 6:

- a) As part of test case PP-20I, configured the TLS Tool to modify a byte in the Server Finished handshake message, and verified that the TOE sent fatal alert message and encrypted message followed by a FIN and ACK message.
- b) As part of test case PP-20J, custom TLS Tool was configured to send a garbled message from the server after the server has issued the ChangeCipherSpec message and verified that the TOE denied the connection.
- c) As part of test case PP-20K, configured TLS Tool to modify at least one byte in the server's nonce in the Server Hello handshake message, and verified that the TOE rejected the Server Key Exchange handshake message.

2.2.9.2 FCS_TLSC_EXT.1.2

2.2.9.2.1 TSS Assurance Activities

TSS Assurance Activities:

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.

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

TSS Implementation Details/Results:

The ST, Section 7.2 describes the client's method of establishing all reference identifiers as DNS names or IPv4 addresses as defined in RFC 3986 in the Common Name or Subject Alternative Name (SAN) of the presented server certificate. The TOE also supports DNS identifiers that include wildcards.

The ST, Section 7.2 confirms that the TOE does not implement certificate pinning and does not support Elliptic Curve Extension in the evaluated configuration.

The ST, Section 7.2 describes that if a server certificate does not have SAN extension but contains IP Address in the Common Name, the TOE performs binary comparison between the presented and reference identifiers.

Assurance Activity Report

2.2.9.2.2 Guidance Assurance Activities

Guidance Assurance Activities:

The evaluator shall ensure that the operational guidance describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE 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.

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.

Guidance Implementation Details/Results:

The evaluator verified the ADMIN and confirmed that the ADMIN contains instructions for setting the reference identifier for the TOE’s own certificate to be used for the purpose of certificate validation in TLS in Section 4 Authentication; Sub-section: Certificate Page 154 of the ADMIN.

The CC-Addendum Section 3 Evaluated configuration, sub-section 12 configure external audit server (syslog) describes all supported identifiers when configuring the external syslog parameters.

2.2.9.2.3 Testing Assurance Activities

Testing Assurance Activities:

- (1) 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.

- (2) 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.
- (3) The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection:
- a) Test 1 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the CN.

Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1.

Assurance Activity Report

- b) Test 2 [conditional]: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, URI). When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the SAN.
- c) Test 3 [conditional]: If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.
- d) Test 4 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, SRV).
- e) Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URIID):
 - 1) [conditional]: The evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g., foo.*.example.com) and verify that the connection fails.
 - 2) [conditional]: The evaluator shall present a server certificate containing a wildcard in the left-most label (e.g., *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g., foo.example.com) and verify that the connection succeeds, if wildcards are supported, or fails if wildcards are not supported. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g., bar.foo.example.com) and verify that the connection fails. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)
- f) Test 6 [conditional]: If IP addresses are supported, 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 an asterisk (*) (e.g. CN=192.168.1.* when connecting to 192.168.1.20, 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).

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 6.

- g) Test 7 [conditional]: If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator modifies each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier):
 - 1) The evaluator shall present a server certificate that does not contain an identifier in the Subject (DN) attribute type(s) that matches the reference identifier. The evaluator shall verify that the connection fails.
 - 2) The evaluator shall present a server certificate that contains a valid identifier as an attribute type other than the expected attribute type (e.g. if the TOE is configured to expect id-serialNumber=correct_identifier, the certificate could instead include id-at-

Assurance Activity Report

name=correct_identifier), and does not contain the SAN extension. The evaluator shall verify that the connection fails. Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass this test.

- 3) The evaluator shall present a server certificate that contains a Subject attribute type that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.
- 4) The evaluator shall confirm that all use of wildcards results in connection failure regardless of whether the wildcards are used in the left or right side of the presented identifier. (Remark: Use of wildcards is not addressed within RFC 5280.)

Testing Implementation Details/Results:

The evaluator configured the reference identifier in the Common Name and Subject Alternative Name extension of the server certificate according to the ADMIN/CC-Addendum and performed the following tests during a TLS connection between the TOE and the Custom TLS Tool as an external audit server:

- a) Test 1: As per the test case PP-21A, the evaluator used an otherwise valid certificate with no SAN extension and no matching IP Address or DNS name of the TLS Tool Server in the Common Name of the Subject DN extension of the server certificate. The evaluator confirmed from the traffic capture and the TOE audit log that that connection was denied by the TOE due to "**certificate subject does not match the configured hostname**" error.
- b) Test 2: As per the test case PP-21B, the evaluator used an otherwise valid server certificate with correct identifier in the Common Name but invalid value in the SAN extension. The evaluator confirmed from the traffic capture and the TOE audit log that that connection was denied by the TOE due to "**certificate subject does not match the configured hostname**" error.
- c) Test 3: As per the test case PP-21C, the evaluator used a valid server certificate with correct identifier in the Common Name but no SAN extension. The evaluator confirmed from the traffic capture and the TOE audit log that that handshake was successful, and a secure connection was established between the TOE and the TLS Tool Server.
- d) Test 4: As per the test case PP-21D, the evaluator used a valid server certificate with an invalid identifier in the Common Name but a valid SAN extension. The evaluator confirmed from the traffic capture and the TOE audit log that that handshake was successful, and a secure connection was established between the TOE and the TLS Tool Server.
- e) Test 5: The evaluator performed the following wildcard tests with each supported type of reference identifier that included a DNS name in CN attribute and SAN extension in the Server certificate and confirmed the following behaviour of the TOE:
 - 1) The evaluator configured the TLS Tool with a server certificate containing a wildcard that was not in the left- most label of the presented identifier (tool.*.lab.local) and confirmed that the TOE rejected the connection. Refer to test case PP-21EO for the full details.
 - 2) The evaluator configured the TLS Tool with server certificate containing wildcard not in leftmost in the CN/SAN extension (two individual certificates) and confirmed that the TOE rejected the connection. As part of the positive testing, the evaluator configured the TLS Tool with server certificate containing wildcard in the leftmost both in the CN and SAN extension (two individual certificates) and confirmed that the TOE accepted the certificate and connection was established successfully. Refer to test case PP-21O for more details.
- f) Test 6: As per the test case PP-21F, the evaluator presented a server certificate that contains a CN that matches the reference identifier, except one of the groups has been replaced with an asterisk (*) (certificate contains CN=192.168.0.* when connecting to 192.168.0.205). The server certificate did not contain the SAN extension. The evaluator observed that the TOE connection to the server failed.
- g) Test 7: does not apply as there is no TLS-based trusted path communications according to FPT_ITT.1.

SonicWALL SMA V12.4

Assurance Activity Report

2.2.9.3 FCS_TLSC_EXT.1.3

2.2.9.3.1 TSS Assurance Activities

TSS Assurance Activities: N/A
TSS Implementation Details/Results: N/A

2.2.9.3.2 Guidance Assurance Activities

Guidance Assurance Activities: N/A
Guidance Implementation Details/Results: N/A

2.2.9.3.3 Testing Assurance Activities

<p>Testing Assurance Activities:</p> <p>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.</p> <p>Test 2: The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g. trusted channel was established) covering the types of failure for which an override mechanism is defined.</p> <p>Test 3: [conditional]: The purpose of this test to verify that only selected certificate validation failures could be administratively overridden. If any override mechanism is defined for failed certificate validation, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e., certificate is rejected), and there is no administrative override available to accept such certificate.</p>
<p>Testing Implementation Details/Results:</p> <p>Test 1: Before initiating connection with the external audit server, the evaluator uploaded the relevant CA Certificate chain to the TOE to validate the Syslog server certificate during the TLS handshake. The evaluator verified and confirmed from the packet capture that the handshake was successful and connection was established between TOE and the Syslog server when the TOE initiated the connection. Refer to test case PP-22A for more details.</p> <p>Test 2: The evaluator then removed one of the CA certificate(s) from the certificate chain of the TOE trusted store, which was required to validate the Syslog server certificate. When the TOE initiated the connection, the evaluator observed that the connection was rejected by the TOE when the presented server certificate was not successfully validated due to the missing chain in the TOE trusted store. Refer to test case PP-22B for more details</p> <p>Test 3: not applicable as the TOE does not implement any override mechanism.</p>

2.2.9.4 FCS_TLSC_EXT.1.4

2.2.9.4.1 TSS Assurance Activities

TSS Assurance Activities:

Assurance Activity Report

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

TSS Implementation Details/Results:

The ST, Section 7.2 states that the TOE does not support Elliptical Curve Extension in the evaluated configuration. Therefore, this activity is not applicable.

2.2.9.4.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Implementation Details/Results:

The TOE does claim and implement support for EC cryptography in the evaluated configuration, however there is no configuration required for the TOE's supported Elliptical Curves extension. Therefore, this activity is not applicable.

2.2.9.4.2 Testing Assurance Activities

Testing Assurance Activities:

Test 1: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension, the evaluator shall configure the server to perform ECDHE or DHE (as applicable) key exchange using each of the TOE's supported curves and/or groups. The evaluator shall verify that the TOE successfully connects to the server.

Testing Implementation Details/Results:

Test 1: as part of the test case PP-23: for each TOE's claimed curve (secp256r1, secp384r1), the evaluator configured the customer TLS server to present such curve and verified that the TOE successfully connects to the customer TLS server. Therefore, this activity is satisfied.

2.2.10 FCS_TLSS_EXT.1 TLS Server Protocol

2.2.10.1 FCS_TLSS_EXT.1.1

2.2.10.1.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The ST, Section 7.2 describes the implementation of the TLS and lists the following ciphersuites:

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA384 as defined in RFC 5289

as the supported ciphersuites.

The evaluator verified that the ciphersuites specified match those listed for this component in Section 6.1.2.

Assurance Activity Report

2.2.10.1.2 Guidance Assurance Activities

Guidance Assurance Activities:

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).

Guidance Implementation Details/Results:

The evaluator examined the CC-Addendum, **Section 3 Evaluated configuration Step# 10: Configure TLS settings; Page 34** and confirmed that the CC-Addendum contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.

2.2.10.1.3 Testing Assurance Activities

Testing Assurance Activities:

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

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

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

b) (Test Intent: The intent of this test is to ensure that the server's TLS implementation immediately makes use of the key exchange and authentication algorithms to:

a) Correctly encrypt (D)TLS Finished message and

b) Encrypt every (D)TLS message after session keys are negotiated.). The evaluator shall use one of the claimed ciphersuites to complete a successful handshake and observe transmission of properly encrypted application data. The evaluator shall verify that no Alert with alert level Fatal (2) messages were sent. The evaluator shall verify that the Finished message (Content type hexadecimal 16 and handshake message type hexadecimal 14) is sent immediately after the server's ChangeCipherSpec (Content type hexadecimal 14) message. The evaluator shall examine the Finished message (encrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 11 22 33 44 55...) and confirm that it does not contain unencrypted data (unencrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 14 00 00 0c...), by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14 for at least one of three test messages. There is a chance that an encrypted Finished message contains a hexadecimal value of '14' at the position where a plaintext Finished message would contain the message type code '14'. If the observed Finished message contains a hexadecimal value of '14' at the position where the plaintext Finished message would contain the message type code, the test shall be repeated three times in total. In case the value of '14' can be observed in all three tests it can be assumed that the Finished message has indeed been sent in plaintext and the test has to be regarded as 'failed'. Otherwise it has to be assumed that the observation of the value '14' has been due to chance and that the Finished message has indeed been sent encrypted. In that latter case the test shall be regarded as 'passed'.

Testing Implementation Details/Results:

SonicWALL SMA V12.4

Assurance Activity Report

Test 1: The TOE implements trusted channel with remote client (browser) to access AMC. In this configuration, the TOE acts as a TLS server. The following TLS ciphers are supported in the evaluated configuration:

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA384 as defined in RFC 5289

In test case PP-18A, the evaluator established a TLS connection using each of the ciphersuites specified by the above requirement as given in the ST. A custom test tool was used to force negotiation of each supported ciphersuite. Packet capture was then used to observe the handshake and to confirm that the selected ciphersuite was successfully negotiated between the TOE and the custom tool.

Test 2: As part of test case PP-18B, the evaluator configured the custom tool to send Client Hello to the TOE TLS server with a list of ciphersuites that does not contain any of the cipher suites in the server's ST and verified that the server denied the connection. Additionally, the evaluator sent a Client Hello to the server using the custom tool containing only the TLS_NULL_WITH_NULL_NULL cipher suite and verified that the server denies the connection.

Test 3: The evaluator used a custom test tool to send modified traffic as specified in the assurance activities. For the following tests, the test tool is acting as a TLS client and the TOE is acting as a server.

- a) In test case PP-18C, the evaluator configured the custom tool to modify a byte in the Client Finished handshake message and verified that the server rejected the connection and did not send any application data.
- b) In test case PP-18D, the evaluator configured the custom tool to send FINISH message instead of CHANGE_CIPHER_SPEC and then tried to resume the previous session and verified that the TOE rejected the connection from the client.

2.2.10.2 FCS_TLSS_EXT.1.2

2.2.10.2.1 TSS Assurance Activities

TSS Assurance Activities:

The evaluator shall verify that the TSS contains a description of the denial of old SSL and TLS versions.

TSS Implementation Details/Results:

The evaluator verified the ST Section 7.2 and confirmed that in the evaluated configuration the TOE supports only TLS v1.2 secure communication protocols that conforms to RFC 5246.

2.2.10.2.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Implementation Details/Results:

The evaluator examined the CC-Addendum, **Section 3 Evaluated configuration Step# 10: Configure TLS settings; Page 34** and confirmed that it contains necessary configuration to configure the TLS v1.2 as described in the TSS section 7.2.

2.2.10.2.3 Testing Assurance Activities

Testing Assurance Activities:

SonicWALL SMA V12.4

Assurance Activity Report

The evaluator shall send a Client Hello requesting a connection for all mandatory and selected protocol versions in the SFR (e.g. by enumeration of protocol versions in a test client) and verify that the server denies the connection for each attempt.

Testing Implementation Details/Results:

As part of test case PP-18E, the evaluator chose the options for deprecated SSL version (SSL 1.0, SSL, 2.0, SSL 3.0, TLS 1.0 and TLS 1.1) in the browser that is used to access AMC and verified that the TOE rejected the connection when the evaluator tried to establish connection between the browser and TOE TLS server, handshake failed due to the deprecated SSL/TLS version.

2.2.10.3 FCS_TLSS_EXT.1.3

2.2.10.3.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The ST section 6.1.2 claims that the ECDHE is the key agreement method. The evaluator reviewed the ST Section 7.2 and noted that it describes the ECDHE key exchange, by saying that the TOE generates EC Diffie-Hellman parameters over NIST curves secp256r1 and secp384r1. The Server Key Exchange Message implements key agreement parameters according to RFC 5246 Section 7.4.3.

2.2.10.3.2 Guidance Assurance Activities(redo)

Guidance Assurance Activities:

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

Guidance Implementation Details/Results:

The evaluator examined the CC-Addendum, **Section 3 Evaluated configuration Step# 10: Configure TLS settings; Page 34** and confirmed that it contains necessary configuration to configure the TLS versions to TLS v1.2 as described in the TSS section 7.2.

2.2.10.3.3 Testing Assurance Activities

Testing Assurance Activities:

Test 1: [conditional] If ECDHE ciphersuites are supported:

- a) The evaluator shall repeat this test for each supported elliptic curve. The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single supported elliptic curve specified in the Elliptic Curves Extension. The Evaluator shall verify (through a packet capture or instrumented client) that the TOE selects the same curve in the Server Key Exchange message and successfully establishes the connection.
- b) The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single unsupported elliptic curve (e.g. secp192r1 (0x13)) specified in RFC4492, chap. 5.1.1. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.

Test 2: [conditional] If DHE ciphersuites are supported, the evaluator shall repeat the following test for each supported parameter size. If any configuration is necessary, the evaluator shall configure the TOE to use a

SonicWALL SMA V12.4

Assurance Activity Report

supported Diffie-Hellman parameter size. The evaluator shall attempt a connection using a supported DHE ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Exchange Message where p Length is consistent with the message are the ones configured Diffie-Hellman parameter size(s).

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.

Testing Implementation Details/Results:

Test 1a: during the test case PP-18G, the evaluator used CygnaCom internal TLS tool and configured to attempt a TLS connection toward the TOE using a single supported ECDHE key agreement, then the evaluator reviewed the network traffic capture and confirmed that the TOE selected the same curve in the Server Key Exchange message and successfully established the TLS connection.

Test 1b: during the test case PP-18F, the evaluator used CygnaCom internal TLS tool and configured to attempt a TLS connection toward the TOE using a supported ECDHE key agreement, then the evaluator reviewed the network traffic capture and confirmed that the TOE does not send a Server Hello message and the connection is not successfully established.

Test 2: This test case is not applicable, as the ST does not claim the DHE key establishment ciphersuites.

Test 3: This test case is not applicable, as the ST does not claim the RSA key establishment ciphersuites.

Assurance Activity Report

2.2.10.4 FCS_TLSS_EXT.1.4

2.2.10.4.1 TSS Assurance Activities

TSS Assurance Activities:

- (1) The evaluator shall verify that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246) and/or if session resumption based on session tickets is supported (RFC 5077).
- (2) If session tickets are supported, the evaluator shall verify that the TSS describes that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption. The evaluator shall verify that the TSS identifies the key lengths and algorithms used to protect session tickets.
- (3) 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.
- (4) If the TOE claims a (D)TLS server capable of session resumption (as a single context, or across multiple

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

Note: TD0569 https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0569 was applied.

TSS Implementation Details/Results:

- (1) In the ST section 6.1.2, the ST author claims support for session tickets according to RFC5077. The ST, TSS section 7.2, it states that the TOE supports session resumption in API context based on session tickets according to RFC 5077.
- (2) The ST, TSS section 7.2, stays that the session tickets adhere to the RFC 5077 ticket structure of NAME [16], IV[16], STATEDATA[varies], HMAC[32], where the hash algorithm is SHA256 and the cipher is AES-256-CBC.
- (3) The ST, TSS section 7.2, it stays that the session tickets adhere to the RFC 5077 ticket structure of NAME [16], IV[16], STATEDATA[varies], HMAC[32].
- (4) The ST, TSS Section 7.2, describes how the session resumption according to RFC 5077, is coordinated across those contexts, by stating that the connection with VPN clients has multiple contexts – probe, API, and Tunnel. Probe is there to establish if mutual authentication is configured and will always result in a failed handshake in the evaluated configuration. API context handles domain authentication and supports renegotiation. Tunnel context is used for transport and is only possible after API context successfully established.

2.2.10.4.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Note: TD0569 https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0569 was applied.

Guidance Assurance Activities Details/Results:

This guidance assurance activity is in regard to the TOE's support for TLS session resumption. The ST does not claim support for TLS session ID. With regards to the session ticket according to RFC 5077, during the testing assurance activity for the resumption test cases PP-181, the evaluator did not have to make any configuration change to enable the TOE's support for TLS session ticket, therefore this guidance assurance activity is considered satisfied.

Assurance Activity Report

2.2.10.4.3 Testing Assurance Activities

Testing Assurance Activities:

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 1 [conditional]: If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test:

- a) The client sends a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket.
- b) The client verifies the server does not send a NewSessionTicket handshake message (at any point in the handshake).
- c) The client verifies the Server Hello message contains a zero-length session identifier or passes the following steps:
Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID.
- d) The client completes the TLS handshake and captures the SessionID from the ServerHello.
- e) The client sends a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d).
- f) f) The client verifies the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

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 2 [conditional]: If the TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

- a) The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246).
- b) The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

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

Assurance Activity Report

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

Test 3 [conditional]: If the TOE supports session tickets according to RFC5077, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

- a) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE responds with an abbreviated handshake described in section 3.1 of RFC 5077 and illustrated with an example in figure 2. Of particular note: if the server successfully verifies the client's ticket, then it may renew the ticket by including a NewSessionTicket handshake message after the ServerHello in the abbreviated handshake (which is shown in figure 2). This is not required, however as further clarified in section 3.3 of RFC 5077).
- b) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then modify the session ticket and send it as part of a new Client Hello message. The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data.

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.

Note: TD0569 (https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?td_id=0569), and TD0556 (https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0556) were applied.

c)

Testing Assurance Activities Details/Results:

Test 1: not applicable, as the TOE support TLS session resumption according to RFC5077.

Test 2: not applicable, as the TOE does not support TLS resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2).

Test 3a: during the test case PP-18I, the evaluator used a customer TLS client to mimic the VPN connect tunnel client and observed the TOE use of session ticket according to RFC 5077, therefore the evaluator considered the testing assurance activity fulfilled.

Test 3b: during the test case PP-18I, the evaluator used a customer TLS client to perform modify the session ticket and send it as part of a new Client Hello message. Using a network capture tool, the evaluator reviewed the captured traffic and confirmed that the TOE implicitly rejected the session ticket by performing a full handshake.

Assurance Activity Report

2.2.11 FCS_TLSS_EXT.2 Extended: TLS Server support for mutual authentication

2.2.11.1 FCS_TLSS_EXT.2.1 and FCS_TLSS_EXT.2.2

2.2.11.1.1 TSS Assurance Activities

TSS Assurance Activities:

- (1) The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.
- (2) The evaluator shall verify the TSS describes how the TSF uses certificates to authenticate the TLS client. The evaluator shall verify the TSS describes if the TSF supports any fallback authentication functions (e.g. username/password, challenge response) the TSF uses to authenticate TLS clients that do not present a certificate. If fallback authentication functions are supported, the evaluator shall verify the TSS describes whether the fallback authentication functions can be disabled.

TSS Implementation Details/Results:

- (1) The evaluator examined the ST, Section 7.7, and confirmed that it describes the use of client-side certificates for the TLS mutual authentication, by saying that “The TOE protects communications with the Connect Tunnel VPN client by establishing a mutually authenticated secure channel between itself and the client. To implement this trusted channel, the TOE uses TLS v1.2 protocol with certificate-based mutual authentication.” and when it says, “The client identifiers are configured via MGMT_CSFC_CERTIFICATE_REQUIRED_ATTRIBUTES CEM parameter where individual RDNs are separated by '&&' (e.g. 'C=US && O=SonicWall Inc.'). If any of the attributes do not match the configured values the client certificate will be rejected. Trusted CAs have to be imported into the TOE and manually added to the TOE’s trust store. If the chain of trust cannot be established the client certificate will be rejected.
- (2) The evaluator examined the ST, Section 7.7, and confirmed that it describes if the TSF supports any fallback authentication, when it states that “If connecting client fails to present a certificate, the connection is rejected with no fallback authentication option”.

2.2.11.1.2 Guidance Assurance Activities

Guidance Assurance Activities:

- (1) If the TSS indicates that mutual authentication using X.509v3 certificates is used, the evaluator shall verify that the AGD guidance includes instructions for configuring the client-side certificates for TLS mutual authentication.
- (2) The evaluator shall verify the guidance describes how to configure the TLS client certificate authentication function. If the TSF supports fallback authentication functions, the evaluator shall verify the guidance provides instructions for configuring the fallback authentication functions. If fallback authentication functions can be disabled, the evaluator shall verify the guidance provides instructions for disabling the fallback authentication functions.

Guidance Assurance Activities Details/Results:

- 1) The evaluator examined the CC-Addendum, Section 5 Configuring TLS certificates on the Client; Page 39 and confirmed that the CC-Addendum contains instructions on configuring the client-side certificates for TLS mutual authentication.
- 2) The evaluator examined the CC-Addendum, Section 6 Client Certificate Validation; Page 40 and confirmed that the CC-Addendum contains instructions on configuring the TLS client certificate authentication function. The TOE does not support a fallback authentication functions.

2.2.11.1.3 Testing Assurance Activities

Testing Assurance Activities:

Assurance Activity Report

Test 1a [conditional]: If the TOE requires or can be configured to require a client certificate, the evaluator shall configure the TOE to require a client certificate and send a Certificate Request to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify that the handshake is not finished successfully and no application data flows.

Test 1b [conditional]: If the TOE supports fallback authentication functions and these functions cannot be disabled. The evaluator shall configure the fallback authentication functions on the TOE and configure the TOE to send a Certificate Request to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify the TOE authenticates the connection using the fallback authentication functions as described in the TSS.

Note: Testing the validity of the client certificate is performed as part of the X.509 testing.

Test 2 [conditional]: If TLS 1.2 is claimed for the TOE, the evaluator shall configure the server to send a certificate request to the client without the supported_signature_algorithm used by the client's certificate. The evaluator shall attempt a connection using the client certificate and verify that the connection is denied.

Test 3: The aim of this test is to check the response of the server when it receives a client identity certificate that is signed by an impostor CA (either Root CA or intermediate CA). To carry out this test the evaluator shall configure the client to send a client identity certificate with an issuer field that identifies a CA recognized by the TOE as a trusted CA, but where the key used for the signature on the client certificate does not correspond to the CA certificate trusted by the TOE (meaning that the client certificate is invalid because its certification path does not terminate in the claimed CA certificate). The evaluator shall verify that the attempted connection is denied.

Test 4: The evaluator shall configure the client to send a certificate with the Client Authentication purpose in the extendedKeyUsage field and verify that the server accepts the attempted connection. The evaluator shall repeat this test without the Client Authentication purpose and shall verify that the server denies the connection. Ideally, the two certificates should be identical except for the Client Authentication purpose.

Test 5: The evaluator shall perform the following modifications to the traffic:

- a) Configure the server to require mutual authentication and then connect to the server with a client configured to send a client certificate that is signed by a Certificate Authority trusted by the TOE. The evaluator shall verify that the server accepts the connection.
- b) Configure the server to require mutual authentication and then modify a byte in the signature block of the client's certificate Verify handshake message (see RFC5246 Sec 7.4.8). The evaluator shall verify that the server rejects the connection.

Note: Testing the validity of the client certificate is performed as part of the X.509 testing.

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

Test 6: 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 7: The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g.

SonicWALL SMA V12.4

Assurance Activity Report

trusted channel was established) covering the types of failure for which an override mechanism is defined.

Test 8 [conditional]: The purpose of this test is to verify that only selected certificate validation failures could be administratively overridden. If any override mechanism is defined for failed certificate validation, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative override available to accept such certificate.

Testing Assurance Activities Details/Results:

Test 1a: during test case PP-19A, the evaluator used a customer TLS client and attempted a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator verified that the handshake is not finished successfully, and no application data flows.

Test 1b: according to the ST section 6.1.2, the ST does not claim support for the fallback authentication functions, therefore this testing activity is considered fulfilled.

Test 2: during the test case PP-19B, the evaluator configured a customer TLS client to send the TOE an x509 certificate signed using unsupported signature_algorithm. The evaluator verified that the TOE rejected the TLS client certificate and closed the connection.

Test 3: during the test case PP-19C, the evaluator loaded a custom TLS client with an x509 certificate signed by an impostor CA, the evaluator observed that the TOE rejected the TLS client x509 certificate and closed the TLS connection, therefore this testing assurance activity is considered fulfilled.

2.2.11.2 FCS_TLSS_EXT.2.3

2.2.11.2.1 TSS Assurance Activities

TSS Assurance Activities:

The evaluator shall verify that the TSS describes which types of identifiers are supported during client authentication (e.g. Fully Qualified Domain Name (FQDN)). If FQDNs are supported, the evaluator shall verify that the TSS describes that corresponding identifiers are matched according to RFC6125. For all other types of identifiers, the evaluator shall verify that the TSS describes how these identifiers are parsed from the certificate, what the expected identifiers are and how the parsed identifiers from the certificate are matched against the expected identifiers.

TSS Implementation Details/Results:

The evaluator examined the ST, Section 7.7, and verified that it describes the type of x509 identifier supported by the TOE during the VPN connect tunnel client when it states that the client identifiers are configured via MGMT_CSFC_CERTIFICATE_REQUIRED_ATTRIBUTES CEM parameter where individual RDNs are separated by '&&' (e.g., 'C=US && O=SonicWall Inc.'). If any of the attributes do not match the configured values the client certificate will be rejected.

2.2.11.2.2 Guidance Assurance Activities

Guidance Assurance Activities:

The evaluator shall ensure that the AGD guidance describes the configuration of expected identifier(s) for X.509 certificate-based authentication of TLS clients. The evaluator ensures this description includes all types of identifiers described in the TSS and, if claimed, configuration of the TOE to use a directory server.

Guidance Assurance Activities Details/Results:

SonicWALL SMA V12.4

Assurance Activity Report

The evaluator examined the CC-Addendum, Section 5 Configuring TLS certificates on the Client; Step# 13: Configure TLS Mutual Authentication; Page 29 and confirmed that the CC-Addendum contains instructions on configuring all the type of expected identifiers for X.509 certificate-based authentication of the TLS Client.

2.2.11.2.3 Testing Assurance Activities

Testing Assurance Activities:

The evaluator shall send a client certificate with an identifier that does not match an expected identifier and verify that the server denies the connection.

Testing Assurance Activities Details/Results:

In test case PP-19H, the evaluator configured the VPN client software with a client's certificate with an identifier that does not match an expected identifier and observed that the TOE denied the VPN client connection.

2.3 Identification and Authentication (FIA)

2.3.1 FIA_AFL.1.1 Authentication Failure Management

2.3.1.1 TSS Assurance Activities

TSS Assurance Activities:

- (1) 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.
- (2) 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.
- (3) 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).

TSS Implementation Details/Results:

- (1) The evaluator examined the ST Section 7.3 of the TSS and confirmed that the TSS contains the configuration for administrator to configure the number of unsuccessful authentication attempts within a range of 1 to 127 as well as time allowed before a retry is permitted from 1 to 1440 minutes during which, the authenticating user remote AMC interface is locked out.
- (2) The ST, Section 7.3 states that once locked out of the account, the remote administrator would be required to wait for configured amount of time (1 to 1440 minutes) before attempting to log back into the device.
- (3) The ST, Section 7.3 states that "The TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, by distinguishing between local and remote login attempts." The above statement confirms that the local administrator is not locked out due to authentication failures.

2.3.1.2 Guidance Assurance Activities

Guidance Assurance Activities:

- (1) 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.
- (2) 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.

Guidance Implementation Details/Results:

- (1) The evaluator examined the CC-Addendum, Section 3 Evaluated configuration; Step# 3: Configure Admin account username and password restrictions, lockout; Page 15 and confirmed that the guide contains instructions to disable lockout policy for remote administrator. The guide also confirms that the local administrative access is never locked out.
- (2) The evaluator examined the CC-Addendum, Section 3 Evaluated configuration; Step# 3: Configure Admin account username and password restrictions, lockout; Page 15 and confirmed that the guide contains

SonicWALL SMA V12.4

Assurance Activity Report

instructions for configuring the authentication policy to lockout remote administrator accounts (AMC) after certain number of successive unsuccessful authentication attempts and for a certain time period. The instruction provides direction that after the lockout time period, the administrator can login to the device successfully.

2.3.1.3 Testing Assurance Activities

Testing Assurance Activities:

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

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

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

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

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

Testing Implementation Details/Results:

- (1) As part of the test case PP-15, the evaluator, using the CC-Addendum Section 3 Evaluated configuration Page 13, configured the authentication policy lockout invalid login attempts to 7 times, lockout period of 600 seconds after three consecutive invalid login attempts and verified that the remote administrator (via AMC) was not able to login using the valid credentials in the 600 seconds lockout time.
- (2) As part of the test case PP-15, the evaluator confirmed that the after the defined number of unsuccessful authentication attempts, the administrator was not able to login using the valid credentials in the 600 seconds lockout time through AMC. Once the time elapsed, the administrator was able to login to the TOE successfully without any administrator intervention as defined in the ST. As part of the test case PP-15, the evaluator confirmed that the evaluator could not be able to authenticate to the TOE when tried to login just less than the time period configured in Test 1 and confirmed that the authorization attempt using a valid credential does not result in successful access.

2.3.2 FIA_PMG_EXT.1 Password Management

2.3.2.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

Assurance Activity Report

The evaluator examined the ST, Section 7.3 and determined that it contains the list of the supported special characters which are as following: “!” , “@” , “#” , “\$” , “%” , “^” , “&” , “*” , “(” , “)” , [“_” , “+” , “-” , “=” , “{” , “}” , “[” , “]” , “\” , “.” , “,” , “<” , “>” , “?” , “/””.

2.3.2.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

- a) identifies the characters that may be used in passwords and provides guidance to security administrators on the composition of strong passwords, and
- b) provides instructions on setting the minimum password length and describes the valid minimum password lengths supported.

Guidance Implementation Details/Results:

The evaluator examined the ADMIN guide and confirmed the following:

- a) The evaluator examined the CC-Addendum, **Section 3 Evaluated configuration Step# 1 Create a new local authentication server and configure password policy, Page 21** and confirmed that the section contains password policy details on the characters, numbers, length, symbols that can be used to create a valid administrator/user password.
- b) The evaluator examined the CC-Addendum, **Section 3 Evaluated configuration Step# 1 Create a new local authentication server and configure password policy, Page 13** and confirmed that the section contains instructions on setting the minimum password length and describes the valid minimum password lengths supported.

2.3.2.3 Testing Assurance Activities

Testing Assurance Activities:

The evaluator shall perform the following tests.

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

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

Testing Implementation Details/Results:

Test 1: The TOE supports only password-based authentication. As part of the test case PP-3, the evaluator tested all of the password attributes required by the PP, including negative tests. The evaluator tested the following rules for the password-based authentication:

- Passwords composed of any combination of upper and lower case letters, numbers, and the following special characters: (~ ` ! @ # \$ % ^ & * () _ - + = { } [] \ ; : " ' < , > . ? /) .
- Minimum password length of 4 characters

The evaluator set the minimum password length to 4 characters and observed that the set value took effect. As part of the negative testing, the evaluator tested the minimum length password policy by providing a shorter password during user creation and noted that an error message stating that a minimum of 4 characters were required for passwords was produced. Password length greater than 4 characters and password length equal to 4 characters were also tested in the positive tests.

Assurance Activity Report

2.3.3 FIA_UIA_EXT.1 User Identification and Authentication

2.3.3.1 TSS Assurance Activities

TSS Assurance Activities:

- (1) The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon".
- (2) The evaluator shall examine the TSS to determine that it describes which actions are allowed before user identification and authentication. The description shall cover authentication and identification for local and remote TOE administration.

TSS Implementation Details/Results:

- (1) The evaluator examined the ST Section 7.3 of the TSS and confirmed that the section contains information about the administrator logon process to the TOE. The ST says "For remote administration, implemented as a web-based interface secured with TLS, the TOE configured to authenticate itself with X509 certificates. For both local and remote administration, only username and password-based authentication is supported in the evaluated configuration. Upon successful authentication, the TOE assigns administratively defined role to that user for the duration of the session. Successful login is indicated by TOE offering a home page or a command line prompt."
- (2) The evaluator examined The ST Section 7.3 and confirmed that the section covers the details on the identification and authentication of the user by the following statement in the ST "For both local and remote administration, only username and password-based authentication is supported in the evaluated configuration". Based on the above statement the evaluator confirms that both the local and remote user is identified using the username and authenticated to the TOE using the valid corresponding password for the username entered. Also, the TSS sections states that "The TOE supports the use of X.509v3 certificate as defined by RFC 5280 to authenticate connections with authorized IT entities and to authenticate itself to remote administrators." So, prior to identify and authenticate the remote administrator, the TOE should establish a successful TLS connection with the authorized IT entities by presenting TOE's server certificate along with the certificate chain.

2.3.3.2 Guidance Assurance Activities

Guidance Assurance Activities:

- 1) 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.
- 2) For each supported the login method, the evaluator shall ensure the guidance documentation provides clear instructions for successfully logging on.
- 3) 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.

Guidance Implementation Details/Results:

The TOE supports two types of Login method to access the TOE. Local administration by logging to the console and remote administration via web interface to access the AMC. The evaluator examined the CC-Addendum and confirmed that the guide contains necessary steps preparatory steps that can be followed by the administrator to configure TOE with the server certificate to establish a successful TLS connection with the AMC before logging to the AMC interface using the username and password. The evaluator examined the CC-Addendum, **Section 3**

Assurance Activity Report

Evaluated configuration Step# 9 Configure SMA web server certificate, Page 21 and confirmed contains sufficient instructions to create SSL certificate on the TOE for the AMC access.

2.3.3.3 Testing Assurance Activities

Testing Assurance Activities:

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

- a) Test 1: The evaluator shall use the guidance documentation to configure the appropriate credential supported for the login method. For that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.
- b) Test 2: The evaluator shall configure the services allowed (if any) according to the guidance documentation, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.
- c) Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.

Testing Implementation Details/Results:

- a) Test 1: The evaluator followed the CC-Addendum and used a valid username and password to successfully login to the TOE. The evaluator used an invalid username and password and observed that access to the TOE was denied. This was carried out for both local (Console CLI) and remote access (AMC). See PP-15 for valid and invalid local access, and PP-7 for valid and invalid remote access.
- b) Test 2: The evaluator observed that for both local (Console CLI) and remote access (AMC) that the list of services available is limited to those specified in the requirement. See PP-6 for more details.
- c) Test 3: The evaluator observed that for local access the login prompt is displayed after establishing a connection, with no other services permitted. Successful login was indicated by a command line prompt, while a failed login returned the operator to the username login prompt. Thus, there are no actions permitted to the user without prior successful identification and authentication to the TOE.

2.3.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

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

2.3.5 FIA_UAU.7 Protected Authentication Feedback

2.3.5.1 TSS Assurance Activities

TSS Assurance Activities: None

TSS Implementation Details/Results: N/A

2.3.5.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Implementation Details/Results:

The evaluator examined the guidance documentation and did not find any necessary preparation steps to ensure passwords are not revealed while a TOE's administrator is entering it for a local login.

Assurance Activity Report

The evaluator confirmed this behavior during the testing assurance activities.

2.3.5.3 Testing Assurance Activities

Testing Assurance Activities:

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.

Testing Implementation Details/Results:

Test 1: The evaluator authenticated locally to the TOE and observed that the password was obscured during entering of authentication information. This behaviour was observed for console login to the TOE using the test case PP-9.

2.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

2.3.6.1 TSS Assurance Activities

TSS Assurance Activities:

- (1) The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e., where the ST is therefore claiming that they are trivially satisfied). It is expected that revocation checking is performed when a certificate is used in an authentication step and when performing trusted updates (if selected). It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected).
- (2) 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.

TSS Implementation Details/Results:

- (1) The evaluator ensured that the TSS describes the validity check and the rules for extendedKeyUsage in ST Section 7.3 as

"The TOE supports the use of X.509v3 certificates as defined by RFC 5280 to authenticate connections with authorized IT entities and to authenticate itself to remote administrators. When certificate-based authentication is used with remote administrators, the TOE presents its server certificate along with a certificate chain. When certificate-based authentication is used with external IT entities, the TOE validates the presented certificate, checks the chain of trust against the TOE's internal trust store, and performs certificate revocation check.

Certificate validation includes path validation (checking CA certificates in the chain), certificate processing (validating the signature, checking keyUsage), and extension processing (checking basicConstraints and extendedKeyUsage extensions).

Verifying the chain of trust includes validating each certificate in the chain, verifying that each CA certificate has basicConstraints flag set to CA:TRUE, verifying that the certificate path terminates with a valid CA certificate designated as a trust anchor."

- (2) The evaluator ensured from the ST Section 7.3 that the TSS section describes the revocation checking requirement for the successful certificate validation by the following statements "The Revocation

SonicWALL SMA V12.4

Assurance Activity Report

checking is implemented using OCSP and is performed on the intermediate CA and leaf certificates. Regardless of a full chain or leaf/identity certificates being presented to the TOE, revocation is performed on the full chain up to a trust anchor as long as the TOE has all necessary CA certificates to determine the trust. Otherwise, the certificate is rejected as untrusted at an early stage of the certificate validation process.”

If any of these steps fail, the connection is terminated at the handshake stage.

From the above statements regarding the certificate validation and revocation checking, the evaluator confirmed that this activity requirement is satisfied.

2.3.6.2 Guidance Assurance Activities

Guidance Assurance Activities:

- 1) The evaluator shall also ensure that the guidance documentation describes where the check of validity of the certificates takes place,
- 2) 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)
- 3) and describes how certificate revocation checking is performed and on which certificate.

Guidance Implementation Details/Results:

- 1) The evaluator examined the CC-Addendum, Section 6 Client Certificate Validation, Page 40 and confirmed that the section describes how the TOE validate an x509 certificate presented during a secure communication with a third-part IT host.
- 2) The evaluator examined the CC-Addendum, Section 5 Configuring TLS certificates on the Client; Step# 13: Configure TLS Mutual Authentication; Page 29 and confirmed that it describes the different rules for the certificate attributes.
- 3) The evaluator examined the ADMIN, Section 4 Authentication, Sub-sections Certificates, topic CA certificates and found the section configuring client certificate revocation, describes how certificate revocation checking is performed and on which client certificate.

2.3.6.3 Testing Assurance Activities

Testing Assurance Activities:

The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected).

The evaluator shall perform the following tests for FIA_X509_EXT.1.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols:

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 TOE's trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store).

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

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

Test 3: The evaluator shall test that the TOE can properly handle revoked certificates—conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the peer certificate and revocation of the peer intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore the revoked certificate(s) used for testing shall not be a trust anchor.

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

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

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

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

Assurance Activity Report

the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

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

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

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 confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e., when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

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

Note: TD0527 (https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0527) was applied.

Testing Implementation Details/Results:

For the following test cases, two-tier hierarchical CA structure and a single-tier CA were used:

- **Root CA → Intermediate CA1 → end entity certificate** for positive testing
- **Untrusted CA → end entity certificate** for negative testing

The TOE supports OCSP Online Certificate Status Protocol for certificate revocation checking. OpenSSL based OCSP responder was installed to provide OCSP service to both root and intermediate CAs.

SonicWALL SMA V12.4

Assurance Activity Report

The evaluator created an SSL profile on the TOE (as per the CC-Addendum Section 3 Evaluated configuration) and created a TOE Server Certificate and imported the respective CA certificates to the TOE trusted store to form the successful chain from the TOE certificate to the Root Certificate.

FIA_X509_EXT.1.1/Rev:

- Test 1a: As part of PP-14A, the evaluator confirmed that an **Untrusted CA** certificate was loaded into the TOE's trust store. The evaluator then used an audit server to present a valid 2048-bit RSA X509v3 leaf certificate signed by an Untrusted CA and observed that the handshake succeeded.
- Test 1b: The evaluator then removed the Untrusted CA, from the TOE's trust store and observed that the validation of the audit server certificate failed due to the absence of the CA certificate.
- Test 2: For this test, TLS Custom Tool was used as the syslog server and TOE was used as a syslog client. As part of PP-14B, the evaluator used a valid, but expired 2048-bit RSA X509v3 server certificate signed by IntCA1 during the TOE's client connection to the TLSTool server. The evaluator observed that the TLS handshake failed, and the TOE logged "**TLS connection failed – unable to establish connection: certificate expired**" message.
- Test 3: The TOE implements OCSP protocol to perform revocation checking. As part of PP-10C, the evaluator used TLS handshake to present a structurally valid 2048-bit RSA X509v3 leaf certificate signed by IntCA1 that was revoked and observed that the handshake failed. The evaluator repeated this test by revoking IntCA1 and observed that the handshake failed.
- Test 4: The TOE implements OCSP protocol to perform revocation checking. As part of PP-14D, the evaluator created a new OCSP responder certificate signed by the intCA1, but the OCSP responder certificate did not contain OCSP Signing attribute in the extendedKeyUsage extension of the certificate. The evaluator observed that during the TLS handshake the TOE rejected the server certificate as it could not validate the OCSP responder certificate. This behaviour satisfied this activity requirement.
- Test 5: As part of PP-14E, the evaluator configured a TLS server to present an otherwise valid RSA X509v3 certificate that contained a modified byte in the first eight bytes of the certificate. The evaluator observed TLS handshake and confirmed that the certificate validation failed, and the TOE produced: "**invalid certificate error**".
- Test 6: As part of PP-14F, the evaluator configured a TLS server to present an otherwise valid RSA X509v3 certificate that contained a modified byte in the last eight bytes of the certificate and observed that the certificate validation failed. The evaluator observed TLS handshake and the TOE produced: "**invalid certificate error**".
- Test 7: As part of PP-14G, the evaluator configured a TLS server to present an otherwise valid RSA X509v3 certificate that contained a modified byte in the public key of the certificate. The evaluator observed TLS handshake and confirmed that the certificate validation failed, and the TOE produced: "**TLS connection failed – unable to establish connection: bad signature**".
- Test 8a and 8b:** not applicable, as the TOE only accepts the CAs chain.
- Test 8c:** As part of PP-14J, the evaluator used an intermediate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator loaded the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator then used an intermediate 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 loaded the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store.

FIA_X509_EXT.1.2/Rev:

Assurance Activity Report

Test 1: As part of PP-15A, the evaluator attempted to install an intermediate CA certificate that was missing the basicConstraints extension. The evaluator observed that this certificate failed the format check by the TOE and an error was generated. The certificate was not installed into the TOE.

Test 2: As part of PP-15B, the evaluator attempted to install an intermediate CA certificate that had the CA flag in the basicConstraints extension set the FALSE. The evaluator observed that this certificate failed the format check by the TOE and an error was generated.

2.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication

2.3.7.1 TSS Assurance Activities

TSS Assurance Activities:

(1) 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.

(2) The evaluator shall examine the TSS to confirm that it describes the behaviour of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the guidance documentation contains instructions on how this configuration action is performed.

TSS Implementation Details/Results:

(1) The ST, Section 7.3 details that the TOE requires an X.509v3 certificate to authenticate its management interface (AMC) and support secure TLS connections with remote administrators. The TSS describes that the TOE server certificate for the TLS communication can only be configured and accessed by the Security Administrators.

(2) The evaluator examined the Section 7.3 TSS and confirmed that the TOE supports X.509v3 certificate-based authentication for the remote administrators. When the certificate-based authentication is used with the external IT entities, the TOE validates the presented certificate, checks the chain of trust against the TOE's internal trust store and performs certificate revocation check. If any of the validation fails, the connection is terminated at the handshake stage.

2.3.7.2 Guidance Assurance Activities

Guidance Assurance Activities:

The evaluator shall also ensure that the guidance documentation describes the configuration required in the operating environment so the TOE can use the certificates. The guidance documentation shall also include any required configuration on the TOE to use the certificates. The guidance document shall also describe the steps for the Security Administrator to follow if the connection cannot be established during the validity check of a certificate used in establishing a trusted channel.

Guidance Implementation Details/Results:

The ADMIN, Section Certificate Strategy, page 147, describes how to generate a certificate signing request and details the steps that can be followed to get the CSR signed and installed into the TOE.

The ADMIN, Section Certificate Usage, page 146, details the TOE configuration to use the loaded certificates.

The ST, does not claim usage of any override mechanism in case the OCSP responder is not available or cannot be reachable in the TOE's network.

Assurance Activity Report

2.3.7.3 Testing Assurance Activities

Testing Assurance Activities:

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

- (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.
- (2) 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.
- (3) 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.

Testing Implementation Details/Results:

In order to verify a certificate, the TOE first performs a validity check on the format of the certificate, then checks trust against its internal certificate store, then performs revocation checking on all of the certificates in the chain using OCSP.

- (1) In test case PP-10A, the evaluator configured the TOE to send audit events to an external syslog server through a TLS secure connection and observer syslog authenticating with X.509v3 certificate and the TOE performing revocation checking. Throughout test case, PP-11A the evaluator manipulated the environment, so the TOE was unable to verify validity, trust, or revocation status of the certificate and observed that appropriate responses.
- (2) The TOE does not support administrator-configurable behaviors in case OCSP responder is unavailable. The evaluator did not carry any testing activity and considered the assurance activity as satisfied.

2.3.8 FIA_X509_EXT.3 Extended: Certificate Requests

2.3.8.1 TSS Assurance Activities

TSS Assurance Activities:

If the ST author selects "device-specific information", the evaluator shall verify that the TSS contains a description of the device-specific fields used in certificate requests.

TSS Implementation Details/Results:

The ST author has not selected "device-specific information"; therefore, this activity is not applicable.

2.3.8.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Implementation Details/Results:

The evaluator examined the CC-Addendum, **Section 3 Evaluated configuration Step# 9 Configure SMA web server certificate, Page 32** and confirmed it contains instructions on requesting certificates from a CA, including generation of a Certificate Request Message.

The evaluator examined the CC-Addendum, **Section 3 Evaluated configuration Step# 9 Configure SMA web server certificate, Page 32** and confirmed it contains instructions for establishing <Common Name, Organization, Organizational Unit, or Country> prior to creating the certificate request message.

Assurance Activity Report

2.3.8.3 Testing Assurance Activities

<p>Testing Assurance Activities:</p> <p>The evaluator shall perform the following tests:</p> <ul style="list-style-type: none">a) Test 1: The evaluator shall use the guidance documentation to cause the TOE to generate a Certification Request. The evaluator shall capture the generated message and ensure that it conforms to the format specified. The evaluator shall confirm that the Certification Request provides the public key and other required information, including any necessary user-input information.b) Test 2: The evaluator shall demonstrate that validating a response message to a Certification Request without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the certificate response message and demonstrate that the function succeeds.
<p>Testing Implementation Details/Results:</p> <p>To generate a valid X509 certificate the TOE first generates RSA key pair, then uses the public key to produce a Certificate Signing Request (CSR). Once the CSR is signed by the CA (intermediate CA1 in all test cases), the resulting certificate is imported back into the TOE and used as the TOE's X.509v3 certificate.</p> <ul style="list-style-type: none">a) Test 1: As part of test case PP-17A, the evaluator followed guidance to generate a CSR. The CSR was analyzed to ensure that it conformed to the RFC 2986 and included the TOE's public key and other relevant information.b) Test 2: As part of test case PP-17B, the evaluator created a test scenario where the TOE attempted to validate a signed certificate without a valid certification path (i.e., intCA1 was missing from the TOE), which resulted in a failure. The evaluator then loaded intCA1's certificate containing the verification public key that signed the certificate being questioned and observed that the certificate validation then succeeded.

2.4 Security Management (FMT)

2.4.1 FMT_MOF.1/ManualUpdate Management of Security Functions Behavior

2.4.1.1 TSS Assurance Activities

<p>TSS Assurance Activities: None</p>
<p>TSS Implementation Details/Results: N/A</p>

2.4.1.2 Guidance Assurance Activities

<p>Guidance Assurance Activities:</p> <ul style="list-style-type: none">(1) The evaluator shall examine the guidance documentation to determine that any necessary steps to perform manual update are described.(2) The guidance documentation shall also provide warnings regarding functions that may cease to operate during the update (if applicable).
<p>Guidance Implementation Details/Results:</p> <ul style="list-style-type: none">(1) The evaluator examined the ADMIN guide and confirmed that the Section 7 System Administration; Sub-section: Installing System Update (Page 325) contains instructions to perform manual updates.(2) The evaluator confirmed that the Section 7 System Administration; Sub-section: Rolling back to a Previous Version section contains instructions to roll back to a known state if the update encounters any problem or any problem experienced after installing an upgrade or hotfix.

Assurance Activity Report

2.4.1.3 Testing Assurance Activities

Testing Assurance Activities:

- (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.
- (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.

Testing Implementation Details/Results:

- (1) As part of the test case PP-2A, the evaluator attempted to perform an update using a legitimate update image by authenticating as a user with no administrator privileges. The evaluator observed that the “**Maintenance**” option under “**System Configuration**” was not visible and also the “**Update**” link under “**System Information** → **Version**” section. This confirms that the TOE does not allow a user without administrator privileges to perform image updates to the TOE.
- (2) As part of the test case PP-2A, the evaluator performed the update, with prior authentication as security administrator, using a legitimate update image. The evaluator observed that the menu items are available and accessible by the Security Administrator to execute the update successfully and the update process generated appropriate audit record(s).

2.4.2 FMT_MTD.1/CoreData Management of TSF Data

2.4.2.1 TSS Assurance Activities

TSS Assurance Activities:

- (1) The evaluator shall examine the TSS to determine that, for each administrative function identified in the guidance documentation; those that are accessible through an interface prior to administrator log-in are identified.
- (2) For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.
- (3) 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.

TSS Implementation Details/Results:

- (1) The ST Section 7.4, details that the TOE requires all users to be successfully be identified and authenticated before permitting any TSF-mediated actions.
- (2) The ST Section 7.4, details that local or remote user is not allowed to perform the TSF management functions without successful authentication and authorization.
- (3) The ST Section 7.4, details that only authenticated and authorized users (local or remote) can manage the TOE’s trust store and import X.509v3 certificates to the TOE.

Assurance Activity Report

2.4.2.2 Guidance Assurance Activities

Guidance Assurance Activities:

- (1) 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.
- (2) 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.

Guidance Implementation Details/Results:

- (1) The evaluator examined the CC-Addendum Section 2 and determined that the instructions provided satisfy each of the TSF-data-manipulating functions implemented in response to the requirements of the cPP and confirmed that the Section 3 Evaluated configuration Page 20 provides instructions to ensure that only administrators have access to the functions.
- (2) The evaluator examined the CC-Addendum, Section 3 Evaluated configuration, Page 20 and confirmed that the section contains instructions for the administrator to configure and maintain the trust store in a secure way. As the TOE supports loading of CA certificates, the evaluator reviewed the Section Configure SMA web server certificate and confirmed that it provides sufficient information for the administrator to securely load CA certificate into the trust store.

2.4.2.3 Testing Assurance Activities

Testing Assurance Activities:

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

Testing Implementation Details/Results: N/A

2.4.3 FMT_MTD.1/CryptoKeys Management of TSF Data

2.4.3.1 TSS Assurance Activities

TSS Assurance Activities:

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.

TSS Implementation Details/Results: N/A

2.4.3.2 Guidance Assurance Activities

Guidance Assurance Activities:

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation lists the keys the Security Administrator is able to manage to include the options available (e.g., generating keys, importing keys, modifying keys or deleting keys) and how that how those operations are performed.

Guidance Implementation Details/Results:

The evaluator examined the ADMIN section Exporting and Importing FIPS-Compliant Certificates, and found it describes how a TOE administrator can import externally generated cryptographic keys.

The evaluator examined the ADMIN Part 4 Authentication, Section Network and Authentication Configuration, sub-section Certificates, Sub-Section Server Certificates, Sub-Section Obtaining a Certificate from a Commercial CR,

Assurance Activity Report

Step 1 Generating a Certificate Signing request, and found it describes how a TOE administrator can generate cryptographic keys.

2.4.3.3 Testing Assurance Activities

Testing Assurance Activities:

- (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.
- (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.

Testing Implementation Details/Results:

- (1) As part of the test case PP-13, the evaluator logged into the AMC using the **System Admin** credentials. The evaluator could not be able to execute any X.509 functions either because the options are disabled or not displayed. This confirms that the TOE does not allow a user without administrator privileges to perform any crypto key operations (modify, delete, generate/import).
- (2) As part of the test case PP-9, PP-11, PP-10X and PP-12X, the evaluator was able to perform crypto key operations (modify, delete, generate/import) when logged as a Security Administrator.

2.4.4 FMT_SMF.1 Specification of Management Functions

2.4.4.1 TSS Assurance Activities

TSS Assurance Activities:

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

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

TSS Implementation Details/Results:

The ST, Section 7.4 says "The TOE is designed to be primarily managed via web-based AMC interface that offers all management functions through a GUI. The CLI is a command-line interface restricted to a limited subset of management functionality aimed at initial configuration (Setup Tool) and system status monitoring. The CLI permits authorized administrators to set system time, verify audit logs, and restart appliance".

The remote management interface (AMC) allows the Security Administrator to perform the following TSF management functions:

- Administer the TOE remotely
- Create the TOE access banner

SonicWALL SMA V12.4

Assurance Activity Report

- Set the session inactivity timeout values
- Verify and manually install firmware updates (verification using published hash and digital signature)
- Configure failed login threshold and lockout period
- Generate, import, delete and configure cryptographic keys required by TLS
- Specify ciphersuites for TLS
- Set system time
- Import x.509v3 certificates in trust store
- Verify audit logs
- User initiated session termination
- Appliance Restart

The evaluator using the ADMIN and CC-Addendum Section 2, configured the TOE for the testing activities through PP-1X to PP-8X and confirmed that the TOE satisfies most of the management functions specified in FMT_SMF.1 via remote administrator interface and some via local interface.

The ST Section 7.4 Security Management describes the local administrative interface. The CC-Addendum, Section 2 Common Criteria Configuration; Sub-section: Initial access and network configuration, Page 6, describes the configuration of local and web-based administrative interface to connect to the TOE.

2.4.4.2 Guidance Assurance Activities

Guidance Assurance Activities:

- 1) The evaluator shall examine the TSS, Guidance Documentation and the TOE as observed during all other testing and shall confirm that the management functions specified in FMT_SMF.1 are provided by the TOE.
- 2) The evaluator shall confirm that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface).
- 3) The evaluator shall examine the TSS and Guidance Documentation to verify they both describe the local administrative interface. The evaluator shall ensure the Guidance Documentation includes appropriate warnings for the administrator to ensure the interface is local.

Guidance Implementation Details/Results:

- 1) The evaluator examined the ST section 7.4 Security Management, the CC-Addendum Section 2 and observed the TOE during all other testing assurance activities and confirmed that the management functions as specified in FMT_SMF.1 are provided by the TOE.
- 2) The ST Section 7.4 Security Management, states that the TOE is designed to be primarily managed via web-based AMC interface that offers all management functions through a GUI. The CLI is a command-line interface restricted to a limited subset of management functionality aimed at initial configuration (Setup Tool) and system status monitoring. The CLI permits authorized administrators to set system time, verify audit logs, and restart appliance.
- 3) The ST Section 7.4 Security Management describes the local administrative interface. The CC-Addendum, Section 2 Common Criteria Configuration; Sub-section: Initial access and network configuration, Page 6, describes the configuration of local and web-based administrative interface to connect to the TOE.

2.4.4.3 Testing Assurance Activities

Testing Assurance Activities:

The evaluator tests management functions as part of testing the SFRs identified in Section 3.4. 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.

Testing Implementation Details/Results:

Please refer to the corresponding AA sections according to the list above.

Function	Testing Reference
----------	-------------------

SonicWALL SMA V12.4

Assurance Activity Report

Administer the TOE locally and remotely	PP-7
Create the TOE access banner	PP-7
Set the session inactivity timeout values	PP-6A
User initiated session termination	PP-6B
Verify and manually install firmware updates (verification using published hash)	PP-2A
Configure failed login threshold and lockout period	PP-24
Generate, import, delete and configure cryptographic keys required by TLS	PP-9, PP-10A to 10I, PP-11A, PP-12A and PP-12B
Specify ciphersuites for TLS	PP-13A to PP-13I and PP-14A to PP-14P
Set system time	PP-4
Import x.509v3 certificates in trust store	PP-14A

2.4.5 FMT_SMR.2 Restrictions on Security Roles

2.4.5.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The evaluator confirmed that the ST Section 7.4 details what are the TOE's supported roles and the restrictions of the roles involving administration of the TOE.

2.4.5.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Implementation Details/Results:

The CC-Addendum, Section 2 Common Criteria Configuration; Sub-section: Initial access and network configuration, Page 6, describes the configuration of local and web-based administrative interface to connect to the TOE.

Assurance Activity Report

2.4.5.3 Testing Assurance Activities

<p>Testing Assurance Activities:</p> <ol style="list-style-type: none">(1) 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.(2) The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this cPP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.
<p>Testing Implementation Details/Results:</p> <ol style="list-style-type: none">(1) The evaluator followed the CC-Addendum to configure the local interface and remote interface for both local and remote administration of the TOE. The TOE administered most of the test cases via AMC, which is expected to be the main interface for the Security Administrators.(2) Throughout the testing, the evaluator followed the ADMIN and CC-Addendum to configure the TOE via AMC (http over TLS) for remote administration of the TOE and CLI over console cable for local administration of the TOE.

2.5 Protection of the TSF (FPT)

2.5.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading all symmetric keys)

2.5.1.1 TSS Assurance Activities

<p>TSS Assurance Activities:</p> <p>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.</p>
<p>TSS Implementation Details/Results:</p> <p>The evaluator confirmed that the ST Section 7.2 Table 16 lists all of the TOE's Critical Security Parameters (CSPs) and details how they are protected and stored. The evaluator checked the Section 7.5 and confirmed that the TSS states that "The TOE protects critical security parameters (CSP) such as stored passwords and cryptographic keys, so they are not directly accessible via normal administrative interfaces. Locally stored password information is obscured by use of hashing (SHA256). Additionally, when login-related configuration information is accessed through local TOE interfaces it is obfuscated by representing input with a series of asterisks."</p>

2.5.1.2 Guidance Assurance Activities

<p>Guidance Assurance Activities: None</p>
<p>Guidance Implementation Details/Results: N/A</p>

2.5.1.3 Testing Assurance Activities

<p>Testing Assurance Activities: None</p>
<p>Testing Implementation Details/Results: N/A</p>

Assurance Activity Report

2.5.2 FPT_APW_EXT.1 Protection of Administrator Passwords

2.5.2.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The evaluator confirmed that the ST Section 7.5 describes how credentials are stored and protected. Based on this section of the ST, it is clear that raw password authentication data are not stored in non-volatile memory in the plain text.

2.5.2.2 Guidance Assurance Activities

Guidance Assurance Activities: None

Guidance Implementation Details/Results: N/A

2.5.2.3 Testing Assurance Activities

Testing Assurance Activities: None

Testing Implementation Details/Results: N/A

2.5.3 FPT_TST_EXT.1 TSF Testing

2.5.3.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The evaluator confirmed that the ST Section 7.5 contains details that the TSF performs diagnostics self-test during start-up and generates audit records to document failure. The ST Section 7.5 describes a standard set of cryptographic self-tests that are consistent with industry's best practices. Therefore, the evaluator considers them to be sufficient.

2.5.3.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Implementation Details/Results:

Assurance Activity Report

The ADMIN guide Section 7 System Administration; Sub-section: FIPS Violations Page 333 contains detailed information on the FIPS mode and the integrity validation of the appliance in several ways. This section contains details on the different self-tests that are performed at each power-cycle to verify all FIPS approved cryptographic algorithms are functioning properly, functioning of the random number generators, integrity of the critical binaries, firmware upgrade files and configuration files. It also mentions that If any of these self-tests fail, a message detailing the specific failure will be displayed on the serial console and logged in `/var/log/aventail/fips.log`, and the appliance will be immediately power cycled via a reboot in order to perform the rigorous self-tests for system integrity. Based on the above details, the evaluator considers that the requirement of this activity is met.

2.5.3.3 Testing Assurance Activities

Testing Assurance Activities:

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

- a) Verification of the integrity of the firmware and executable software of the TOE
- b) Verification of the correct operation of the cryptographic functions necessary to fulfil any of the SFRs.

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

- a) [FIPS 140-2], chap. 4.9.1, Software/firmware integrity test for the verification of the integrity of the firmware and executable software. Note that the testing is not restricted to the cryptographic functions of the TOE.
- b) [FIPS 140-2], chap. 4.9.1, Cryptographic algorithm test for the verification of the correct operation of cryptographic functions. Alternatively, national requirements of any CCRA member state for the security evaluation of cryptographic functions should be considered as appropriate.

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

Testing Implementation Details/Results:

- a) The TOE must be configured to be in FIPS mode of operation as part of the evaluated configuration. The evaluator observed visually via output to the console that the integrity tests were executed on startup of the TOE.
- b) The evaluator observed the self-test being executed during the formal testing. When the device starts up, the self-tests are executed as expected.

Assurance Activity Report

2.5.4 FPT_TUD_EXT.1 Trusted Update

2.5.4.1 TSS Assurance Activities

TSS Assurance Activities:

- (1) The evaluator shall verify that the TSS describe how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the TSS needs to describe how and when the inactive version becomes active. The evaluator shall verify this description.
- (2) 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).
- (3) The evaluator shall verify that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails. Alternatively, an approach using a published hash can be used. In this case the TSS shall detail this mechanism instead of the digital signature verification mechanism. The evaluator shall verify that the TSS describes the method by which the digital signature or published hash is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature or published hash of the update, and the actions that take place for both successful and unsuccessful signature verification or published hash verification.
- (4) 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.
- (5) If a published hash is used to protect the trusted update mechanism, then the evaluator shall verify that the trusted update mechanism does involve an active authorization step of the Security Administrator, and that download of the published hash value, hash comparison and update is not a fully automated process involving no active authorization by the Security Administrator. In particular, authentication as Security Administration according to FMT_MOF.1/ManualUpdate needs to be part of the update process when using published hashes.

TSS Implementation Details/Results:

The evaluator verified the ST Section 7.5 and confirmed that:

- (1) The version of the current system software and the product serial number are displayed at the bottom of the left-hand navigation bar on every page in the remote administrative interface (AMC).
- (2) It details TSF software updates. When software updates are available, an administrator may obtain and apply the updates, query the currently active version and when to stop the installation.
- (3) The TOE supports both manual and automatic signature verification on the binary file. A Security Administrator can manually verify the integrity of the binary file using published hash comparison or the TOE also performs verification of the signature contained in the binary file and returns an error message if the verification fails.
- (4) Automatic checking for updates is not selected for FPT_TUD_EXT.1.2 and hence this activity is not applicable.
- (5) The ST states that there is a manual hash verification and confirmation step involved in the update mechanism. The Security Administrator must authenticate to the secure support website where the software downloads are available. The downloaded image must be then transferred to the appliance using an administrative interface (AMC) after the successful authentication to the TOE.

Assurance Activity Report

2.5.4.2 Guidance Assurance Activities

Guidance Assurance Activities:

- (1) 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.
- (2) The evaluator shall verify that the guidance documentation describes how the verification of the authenticity of the update is performed (digital signature verification or verification of published hash). The description shall include the procedures for successful and unsuccessful verification. The description shall correspond to the description in the TSS.
- (3) If a published hash is used to protect the trusted update mechanism, the evaluator shall verify that the guidance documentation describes how the Security Administrator can obtain authentic published hash values for the updates.

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

Guidance Implementation Details/Results:

- (1) The evaluator verified the ADMIN guide and confirmed that Section 7 System Administration; Sub-section: Upgrading, rolling back, or Resetting the System Page 324 contains instructions to query the current software and firmware version of the TOE. The TOE doesn't support delayed activation.
 - (2) The evaluator verified the ADMIN guide and confirmed that Section 7 System Administration; Sub-section: FIPS Violations contains details about the authenticity of the firmware upgrade file and the upgrade process.
- The evaluator verified the ADMIN guide and confirmed that Appendix B Troubleshooting; Sub-section: Verify a downloaded upgrade file contains details about how the Security Administrators can get the published hash for the updates and use for the comparison with the value derived on the appliance.

2.5.4.3 Testing Assurance Activities

Assurance Activity Report

Testing Assurance Activities:

The evaluator shall perform the following tests:

- a) Test 1: The evaluator performs the version verification activity to determine the current version of the product. If a trusted update can be installed on the TOE with a delayed activation, the evaluator shall also query the most recently installed version (for this test the TOE shall be in a state where these two versions match). The evaluator obtains a legitimate update using procedures described in the guidance documentation and verifies that it is successfully installed on the TOE. For some TOEs loading the update onto the TOE and activation of the update are separate steps ('activation' could be performed e.g., by a distinct activation step or by rebooting the device). In that case the evaluator verifies after loading the update onto the TOE but before activation of the update that the current version of the product did not change but the most recently installed version has changed to the new product version. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update and that current version of the product and most recently installed version match again.
- b) Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator first confirms that no updates are pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator obtains or produces illegitimate updates as defined below and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates:
 - 1) A modified version (e.g., using a hex editor) of a legitimately signed update
 - 2) An image that has not been signed
 - 3) An image signed with an invalid signature (e.g., by using a different key as expected for creating the signature or by manual modification of a legitimate signature)
 - 4) If the TOE allows a delayed activation of updates the TOE must be able to display both the currently executing version and most recently installed version. The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify, that both, current version and most recently installed version, reflect the same version information as prior to the update attempt.
- c) Test 3 [conditional]: If the TOE itself verifies a hash value over an image against a published hash value (i.e., reference value) that has been imported to the TOE from outside such that the TOE itself authorizes the installation of an image to update the TOE, the following test shall be performed (otherwise the test shall be omitted). If the published hash is provided to the TOE by the Security Administrator and the verification of the hash value over the update file(s) against the published hash is performed by the TOE, then the evaluator shall perform the following tests. The evaluator first confirms that no update is pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test.
 - 1) The evaluator obtains or produces an illegitimate update such that the hash of the update does not match the published hash. The evaluator provides the published hash value to the TOE and calculates the hash of the update either on the TOE itself (if that functionality is provided by the TOE), or else outside the TOE. The evaluator confirms that the hash values are different, and attempts to install the update on the TOE, verifying that this fails because of the difference in hash values (and that the failure is logged). Depending on the implementation of the TOE, the TOE might not allow the Security Administrator to even attempt updating the TOE after the verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE.
 - 2) The evaluator uses a legitimate update and tries to perform verification of the hash value without providing the published hash value to the TOE. The evaluator confirmed that this attempt fails. Depending on the implementation of the TOE it might not be possible to attempt the verification of the hash value without providing a hash value to the TOE, e.g., if the hash value needs to be handed over to the TOE as a parameter in a command line message and the syntax check of the command prevents the execution of the command without providing a hash value. In that case the mechanism that prevents the execution of this check shall be tested accordingly, e.g., that the syntax check rejects the command without providing a hash value, and the rejection of the attempt is regarded as sufficient verification of the correct behaviour of the TOE in failing to verify the hash. The evaluator then attempts to install the update on the TOE (in spite of the unsuccessful hash verification) and confirmed that this fails. Depending on the implementation of the TOE, the TOE might not allow to even attempt updating the TOE after the

SonicWALL SMA V12.4

Assurance Activity Report

verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE.

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

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

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

Testing Implementation Details/Results:

- a) Test 1: The evaluator manually verified the hash of the update file against the published hash. Upon validation, the evaluator verified the version of the product via AMC, then installed the update. As part of the update installation, the TOE does integrity checking against the update file and installed the update only after the successful validation. After the update, the evaluator performed 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. Delayed activation is not supported by the TOE. Refer to test case PP-1A for the testing details.
- b) Test 2: The evaluator modified an update using a binary editor and then attempted to install it. The signature verification failed, and the Security Administrator is advised not to proceed further and report the problem to the SonicWall Technical Support. So, the update was not installed. The evaluator observed the same behaviour when he tried to install a none-signed image, or an image signed with an invalid signature. Refer to test case PP-1D for the testing details.
- c) Test 3: (Not applicable) Toe does not verify by itself a hash value over an image against a published hash value.

2.5.5 FPT_STM_EXT.1 TSF Reliable Time Stamps

2.5.5.1 TSS Assurance Activities

TSS Assurance Activities:

The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time, and that it provides a description of how the time is maintained and considered reliable in the context of each of the time related functions.

TSS Implementation Details/Results:

The evaluator examined the ST Section 7.5 and noted the following details that the "TOE implements hardware-based real-time clock managed by an embedded OS, which also controls the exposure of administrative functions. This clock is used to produce reliable timestamps that are available for audit trail generation, synchronization with the operational environment, session inactivity checks, and certificate expiration validation." which satisfies requirement of this activity.

Assurance Activity Report

2.5.5.2 Guidance Assurance Activities

<p>Guidance Assurance Activities:</p> <ul style="list-style-type: none">(1) The evaluator examines the guidance documentation to ensure it instructs the administrator how to set the time.(2) 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.
<p>Guidance Implementation Details/Results:</p> <ul style="list-style-type: none">(1) The evaluator examined the ADMIN guide and confirmed that the Section 7 – System Administration; Sub-section: Configuring Time Settings (Page 289) contains instructions for the Security Administrator on how to set the appliance time.(2) Synchronization of the NTP server is not evaluated and so not applicable.

2.5.5.3 Testing Assurance Activities

<p>Testing Assurance Activities:</p> <p>The evaluator shall perform the following tests:</p> <ul style="list-style-type: none">a) Test 1: If the TOE supports direct setting of the time by the Security Administrator then the evaluator uses the guidance documentation to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.b) Test 2: If the TOE supports the use of an NTP server; the evaluator shall use the guidance documentation to configure the NTP client on the TOE and set up a communication path with the NTP server. The evaluator will observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the guidance documentation. <p>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.</p>
<p>Testing Implementation Details/Results:</p> <ul style="list-style-type: none">a) Test 1: The evaluator used the instructions from ADMIN Section 7 – System Administration; Sub-section: Configuring Time Settings (Page 289) and manually set the time on the TOE. The evaluator verified the audit logs and confirmed that the system time was set correctly. Refer to test case PP-2 for the testing details.b) Test 2: Synchronization of the NTP server is not evaluated and so not tested.

2.6 TOE Access (FPT)

2.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

2.6.1.1 TSS Assurance Activities

<p>TSS Assurance Activities:</p> <p>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.</p>
<p>TSS Implementation Details/Results:</p> <p>The evaluator examined the ST, Section 7.6, which states that the TOE is designed to lock accounts after a number of unsuccessful login attempts. The TOE's minimum lockout value must be configured to a non 0 value to enforce an administrator-defined inactivity timeout after which the inactive session is automatically terminated. Once a session (local or remote) has been terminated, the TOE requires the user to re-authenticate. The administrator can force termination of current session by issuing the logout command exit with CLI or by clicking log out with AMC.</p> <p>Therefore the evaluator considers this assurance activity satisfied.</p>

Assurance Activity Report

2.6.1.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Implementation Details/Results:

The evaluator examined the CC-Addendum and confirmed that the Section 3 Evaluated configuration Page 16 contains instructions to configure the inactivity time period for local administration session termination.

2.6.1.3 Testing Assurance Activities

Testing Assurance Activities:

The evaluator shall perform the following test:

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

Testing Implementation Details/Results:

Test 1: As part of the test case PP-4A, the evaluator followed the CC-Addendum Section 3 Configure Idle Timeout, Page 27 and configured the following values 2, 5, 15 minutes for the inactivity time period. The evaluator configured the login session timeout value to 2, 5, 15 minutes, and observed that the local session was terminated after each of the indicated time out values. The evaluator then observed that re-authentication was required when trying to unlock the session.

2.6.2 FTA_SSL.3 TSF-initiated Termination

2.6.2.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The evaluator examined the ST Section 7.6 and determine that it details the administrative remote session termination by stating that once a session (local or remote) has been terminated, the TOE requires the user to re-authenticate. The administrator can force termination of current session by issuing the logout command exit with CLI or by clicking log out with AMC.

The evaluator examined the ST Section 7.3 and determine that it describes the inactivity time period the TOE supports by stating That the TOE permits an administrator to configure the number of unsuccessful authentication attempts within a range of 1 to 127 as well as time allowed before a retry is permitted from 1 to 1440 minutes during which, the authenticating user is locked out.

2.6.2.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Implementation Details/Results:

Assurance Activity Report

The evaluator examined the CC-Addendum and confirmed that the Section 3 Evaluated configuration Page 18 contains instructions to configure the inactivity time period for remote administration session termination.

2.6.2.3 Testing Assurance Activities

Testing Assurance Activities:

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

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

Testing Implementation Details/Results:

Test 1: As part of the test case PP-4A, the evaluator followed the CC-Addendum Section 3 Configure Idle Timeout, Page 27 and configured the following values 2, 5, 15 minutes for the inactivity time period. The evaluator configured the login session timeout value to 2, 5, 15 minutes, and observed that the TLS session was terminated after each of the indicated time out values. The evaluator then observed that re-authentication was required when trying to unlock the session.

2.6.3 FTA_SSL.4 User-initiated Termination

2.6.3.1 TSS Assurance Activities

TSS Assurance Activities:

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

TSS Implementation Details/Results:

The evaluator examined the ST Section 7.6 and determined that it describes how the local and remote administrative sessions are terminated.

2.6.3.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Implementation Details/Results:

The evaluator examined the CC-Addendum and confirmed that the Section 2.1 Initial Access and network configuration Page 6 and Section 2.3 Accessing SMA Management Console Page 12 contains instructions to terminate remote interactive session.

2.6.3.3 Testing Assurance Activities

Testing Assurance Activities:

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

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

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

Testing Implementation Details/Results:

Test 1: The evaluator established an interactive local session with the TOE. The evaluator followed the CC_Addendum Section 2.1 to log off from the session and observed that the session was terminated.

Test 2: The evaluator established a TLS session with the TOE for remote administration. The evaluator followed

Assurance Activity Report

the user guidance to log off from the TLS session and observed that the session was terminated and closed immediately.

2.6.4 FTA_TAB.1 Default TOE Access Banners

2.6.4.1 TSS Assurance Activities

<p>TSS Assurance Activities:</p> <ul style="list-style-type: none">(1) The evaluator shall check the TSS to ensure that it details each administrative method of access (local and remote) available to the Security Administrator (e.g., serial port, SSH, HTTPS).(2) 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).
<p>TSS Implementation Details/Results:</p> <ul style="list-style-type: none">(1) The evaluator examined the ST Section 7.6 and confirmed that it contains details about each administrative method of access (local and remote) available to the Security Administrator. Local administrator accesses the TOE using the local serial port, while remote administrators access the TOE via AMC.(2) The ST Section 7.6 explains that the TOE displays a Security Administrator-specified advisory notice and consent warning message (banner) when a user initiates an interactive session either locally or remotely.

2.6.4.2 Guidance Assurance Activities

<p>Guidance Assurance Activities:</p> <p>The evaluator shall check the guidance documentation to ensure that it describes how to configure the banner message.</p>
<p>Guidance Implementation Details/Results:</p> <p>The evaluator examined the CC-Addendum and confirmed that the Section 3 Evaluated configuration Page 29 contains adequate instructions to configure the banner message for both local and remote login interfaces.</p>

2.6.4.3 Testing Assurance Activities

<p>Testing Assurance Activities:</p> <p>The evaluator shall also perform the following test:</p> <p>Test 1: The evaluator follows the guidance documentation to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.</p>
<p>Testing Implementation Details/Results:</p> <p>Test 1: As part of the test case PP-5, the evaluator followed the CC-Addendum to configure a notice and consent warning message, then established a session with the TOE and observed that the TOE displayed the notice and consent warning message. The evaluator observed that the TOE displayed the notice and consent warning message for both local and remote methods of access.</p>

2.7 Trusted Path/Channels (FTP)

2.7.1 FTP_ITC.1 Inter-TSF Trusted Channel

2.7.1.1 TSS Assurance Activities

TSS Assurance Activities:

- (1) 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.
- (2) 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.

TSS Implementation Details/Results:

- (1) The evaluator examined the ST Section 7.7 entry for FTP_ITC.1, and noted the following:
For Syslog Server as an authorized entity:
"The TOE protects communications with the audit server by establishing a trusted channel between itself and the audit server. To implement this trusted channel, the TOE uses TLS v1.2 protocol with certificate-based authentication. For certificate-based authentication, presented certificate (x.509v3) is first cryptographically validated, confirmed as issued by a trusted CA, checked for revocation, and then identifiers compared. Trusted CAs have to be imported into the TOE and manually added to the TOE's trust store."
For Web Interface (Remote Administration):
"The TOE protects remote management sessions by establishing a trusted path (secured with TLS) between itself and the administrator connected to a dedicated RJ-45 LAN management port."

(2) The evaluator also confirmed this section contains details on the trusted channel communication between the TOE and the authorized entities (external audit server, Remote Web Interface). The evaluator confirmed that all protocols listed in the TSS are specified and included in the requirements of the ST.

2.7.1.2 Guidance Assurance Activities

Guidance Assurance Activities:

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

Guidance Implementation Details/Results:

The evaluator examined the CC-Addendum Section 3 Evaluated configuration **Step# 10 Configuring TLS Settings Page 34** and confirmed that they contain adequate instructions for establishing the allowed protocols with each authorized IT entity (external Audit Server and AMC). A TLS connection is not an interactive one, so there are no instructions required to recover a TLS connection if it is broken unintentionally.

Assurance Activity Report

2.7.1.3 Testing Assurance Activities

Testing Assurance Activities:

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

The evaluator shall perform the following tests:

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

Test 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.

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

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

The evaluator shall, for each instance where the TOE acts as a client utilizing a secure communication mechanism with a distinct IT entity, physically interrupt the connection of that IT entity for the following durations:

- i) a duration that exceeds the TOE's application layer timeout setting,
- ii) a duration shorter than the application layer timeout but of sufficient length to interrupt the network link layer.

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

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

Testing Implementation Details/Results:

Test 1: The evaluator followed the CC-Addendum and set up communication using the TLS protocol with the syslog server and using the TLS protocol with web browser for AMC access.

Test 2: The evaluator verified that once the TLS information was configured, the communication (TLS handshake) was initiated from the TOE to the audit server. In the remote administration access, the client (browser) initiates the connection to the TOE TLS Server.

Test 3: For communication channel with the syslog server and the browser for AMC access, it was observed that data was not sent in plaintext. Wireshark packet capture was used to verify the data was not in plain text.

Test 4: The evaluator physically interrupted the connection to the audit server on TLS using multiple threshold values (5 minutes, 4 hours and overnight). The evaluator observed that the TLS tunnel was reestablished automatically between the TOE and the remote audit server for 4 hours and overnight downtime and observed that the communications were appropriately protected for all the threshold values. Whereas the TLS connection from the browser to the TOE Server needs re-authentication for the administrator login. The connection will be disconnected as soon as the idle timeout is reached for the browser connection.

Assurance Activity Report

2.7.2 FTP_TRP.1/Admin Trusted Path

2.7.2.1 TSS Assurance Activities

TSS Assurance Activities: (1) The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. (2) 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.
--

TSS Implementation Details/Results: (1) The evaluator examined the ST Section 7.7, which states that the TOE uses TLSv1.2 for remote TOE administration. (2) The evaluator confirmed that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the NDcPP requirement, and are included in the relevant SFRs in the ST.
--

2.7.2.2 Guidance Assurance Activities

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

Guidance Implementation Details/Results: The evaluator examined the CC-Addendum Section 3 Evaluated configuration and confirmed that the section contains adequate instructions for establishing the remote administrative sessions for each supported method (TLS).
--

2.7.2.3 Testing Assurance Activities

Testing Assurance Activities: 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. 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.
--

Testing Implementation Details/Results: Test 1: The evaluator followed the CC-Addendum and ADMIN to set up communication using TLS (v1.2) for remote administration. Test 2: The evaluator verified that once the TLS session was established, the channel data was encrypted between the TOE and the external client (browser).

3 Security Assurance Requirements

The sections below specify Evaluation Activities for the Security Assurance Requirements included in the related cPPs. The Evaluation Activities are an interpretation of the more general CEM assurance requirements as they apply to the specific technology area of the TOE.

Assurance Activity Report

3.1 ASE: Security Target Evaluation

3.1.1 General ASE

Evaluation Activities: (1) When evaluating a Security Target, the evaluator performs the work units as presented in the CEM. In addition, the evaluator ensures the content of the TSS in the ST satisfies the EAs specified in Section 2 (Evaluation Activities for SFRs).
Evaluation Activities Details/Results: (1) The evaluator performed the work units as presented in the CEM. The evaluator ensured the content of the TSS in the ST satisfied the EAs specified in Section 2.

3.2 ADV_FSP.1 Basic Functional Specification

3.2.1 Assurance Activities

Evaluation Activities: (1) <i>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.</i> In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g. audit review or performing updates). Additionally, those interfaces that are identified in the ST, or guidance documentation, as adhering to the security policies (as presented in the SFRs), are also considered security relevant. The intent is that these interfaces will be adequately tested, and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied. The set of TSFI that are provided as evaluation evidence are contained in the Administrative Guidance and User Guidance. (2) <i>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.</i> (3) <i>The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.</i> The evaluator uses the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in Section 2, including the EAs associated with testing of the interfaces. It should be noted that there may be some SFRs that do not have an interface that is explicitly “mapped” to invoke the desired functionality. For example, generating a random bit string, destroying a cryptographic key that is no longer needed, or the TSF failing to a secure state, are capabilities that may be specified in SFRs, but are not invoked by an interface. However, if the evaluator is unable to perform some other required EA because there is insufficient design and interface information, then the evaluator is entitled to conclude that an adequate functional specification has not been provided, and hence that the verdict for the ADV_FSP.1 assurance component is a ‘fail’.
Evaluation Activities Details/Results: As per the NDcPP v2.2 Supporting Document Section 5.2, the documents to be examined for this assurance components in an evaluation are Security Target, AGD Documentation and any required supplementary information required by the cPP: no additional 'functional specification' documentation is necessary to satisfy the EAs. The evaluator used the ST, ADMIN and CC-Addendum to identify the TSFIs of the TOE and found that the documents contain sufficient information on purpose and method of use for each TSFI that is identified as security relevant.

Assurance Activity Report

3.3. AGD: Guidance Documents

3.3.1 AGD_OPE.1 Operational User Guidance

3.3.1.1 Assurance Activities

Evaluation Activities:

- (1) *The evaluator shall ensure the Operational guidance documentation is distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.*
- (2) *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.*
- (3) *The evaluator shall ensure that the Operational guidance contains instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.*
- (4) *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.*
- (5) In addition, the evaluator shall ensure that the following requirements are also met.
 - a) The guidance documentation shall contain instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.
 - b) The documentation must describe the process for verifying updates to the TOE for each method selected for FPT_TUD_EXT.1.3 in the Security Target. The evaluator shall verify that this process includes the following steps:
 - 1) Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).
 - 2) Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes instructions that describe at least one method of validating the hash/digital signature.
 - c) The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The guidance documentation shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.

Note: TD0536 (https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0536) was applied.

Evaluation Activities Details/Results:

- (1) The vendor submitted ADMIN and CC-Addendum for the evaluation. The evaluator used both the documents to configure the TOE in the Common Criteria (CC) mode. The evaluator used the CC-Addendum to configure the TOE in the CC mode and referred ADMIN for in-depth details of the configuration.
- (2) The evaluator followed the Section 2.1 to 2.4 of the CC-Addendum to configure the TOE to be setup in an evaluated configuration. futile
- (3) The evaluator examined the procedures in Section 2 of the CC-Addendum to ensure that the guide includes instructions to successfully install the TOE in each Operational Environment.
- (4) The guide provides administrator login information in Section 2.1 of the CC-Addendum for the first-time setup.
- (5) In addition, the evaluator also confirmed the following criteria's:
 - a) The CC-Addendum, Section 3, Evaluated Configuration, Page 20-26 details how to configure the cryptographic engine in FIPS mode.
 - b) The ADMIN, Section Installing System Update, Page 326 details how to obtain and apply updates to the system and determining if an upgrade was successful/unsuccessful.

The product presents a hash to be manually compared by the administrator to a published hash before loading the update as stated in the ST and found during testing. The TOE also performs verification of the signature contained in the binary file and returns an error message if the verification fails.

Assurance Activity Report

3.3.2 Preparative Procedures

3.3.2.1 Assurance Activities

<p>Evaluation Activities:</p> <ol style="list-style-type: none">(1) The evaluator shall examine the Preparative procedures to ensure they include a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target).(2) 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.(3) The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment.(4) The evaluator shall examine the preparative procedures to ensure they include instructions to manage the security of the TSF as a product and as a component of the larger operational environment.(5) In addition, the evaluator shall ensure that the following requirements are also met. The preparative procedures must<ol style="list-style-type: none">a) include instructions to provide a protected administrative capability; andb) identify TOE passwords that have default values associated with them and instructions shall be provided for how these can be changed.
<p>Evaluation Activities Details/Results:</p> <ol style="list-style-type: none">(1) The vendor submitted ADMIN and CC-Addendum for the evaluation. The evaluator used both the documents to configure the TOE in the Common Criteria (CC) mode. The evaluator used the CC-Addendum to configure the TOE in the CC mode and referred ADMIN for in-depth details of the configuration.(2) The evaluator followed the ADMIN and Section 2 of the CC-Addendum to configure the TOE to be setup in an evaluated configuration.(3) The evaluator examined the procedures in ADMIN and CC-Addendum Section 2 to ensure that the guide includes instructions to successfully install the TOE in each Operational Environment.(4) The guide provides administrator login information in CC-Addendum Section 3 for the first-time setup.(5) In addition, the evaluator also confirmed the following criteria's:<ol style="list-style-type: none">a) The CC-Addendum, Section 2.1, Initial access and network configuration, Page 6 details how to configure a protected administrative capability.b) The CC-Addendum, Section 2.2, Setup Wizard, Page 8 contains the details on the default TOE password and provides instructions on the password change for the administrator.

3.4 ALC: Life-cycle Support

3.4.1 ALC_CMC.1 Labelling of the TOE

3.4.1.1 Assurance Activities

<p>Evaluation Activities:</p> <p>When evaluating that the TOE has been provided and is labelled with a unique reference, the evaluator performs the work units as presented in the CEM.</p>
<p>Evaluation Activities Details/Results:</p> <p>The evaluator performed the CEM work units as reported in the ETR.</p>

3.4.2 ALC_CMS.1 TOE CM coverage

3.4.2.1 Assurance Activities

<p>Evaluation Activities:</p>

Assurance Activity Report

When evaluating the developer's coverage of the TOE in their CM system, the evaluator performs the work units as presented in the CEM.

Evaluation Activities Details/Results:

The evaluator performed the CEM work units as reported in the ETR.

3.5 ATE: Tests

3.5.1 ATE_IND.1 Independent Testing – Conformance

3.5.1.1 Assurance Activities

Evaluation Activities:

The focus of the testing is to confirm that the requirements specified in the SFRs are being met. Additionally, testing is performed to confirm the functionality described in the TSS, as well as the dependencies on the Operational guidance documentation is accurate.

The evaluator performs the CEM work units associated with the ATE_IND.1 SAR. Specific testing requirements and EAs are captured for each SFR in Sections 2, 3 and 4.

The evaluator should consult Appendix B when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

Evaluation Activities Details/Results:

Please refer to the TSTRPT for all details.

3.6 AVA: Vulnerability Assessment

3.6.1 AVA_VAN.1 Vulnerability Survey

3.6.1.1 Assurance Activities

Evaluation Assurance Activities:

- (1) 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 at a minimum the processors used by the TOE. Software components include applications, the operating system and other major components that are independently identifiable and reusable (outside of the TOE), for example a web server, protocol or cryptographic libraries, (independently identifiable and reusable components are not limited to the list provided in the example). This additional documentation is merely a list of the name and version number of the components and will be used by the evaluators in formulating vulnerability hypotheses during their analysis

- (2) The evaluator formulates hypotheses in accordance with process defined in Appendix A. The evaluator documents the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3. The evaluator shall perform vulnerability analysis in accordance with Appendix A.2. The results of the analysis shall be documented in the report according to Appendix A.3.

Note: TD0547 https://www.niap-ccevs.org/Documents_and_Guidance/view_td.cfm?TD=0547 was applied.

Evaluation Activities Details/Results:

- (1) The complete vulnerability analysis is documented in the ETR and in the following reports:

- SonicWall SMA v12.4 Vulnerability Analysis Report Sep 14 2021.xlsx
- SonicWall SMA v12.4 Vulnerability Analysis Report Mar 29 2021.xlsx
- SonicWall SMA v12.4 Vulnerability Analysis Report July 22 2021.xlsx

SonicWALL SMA V12.4

Assurance Activity Report

- Nessus Vulnerability Scan Report Jun 7 2021.html
- Nessus Vulnerability Scan Report March 16-2021.html

The evaluator performed 2 CVE searches on March 16, July 08, and Sep 14, 2021 according to 248 search terms that included the TOE and all the internal components that compose the TOE. The search identified 753 results from which 74 CVEs were deemed as potentially applicable vulnerability.

When a CVE search produced a search result, the evaluator examined CVE details to determine if it is applicable to the TOE in the evaluated configuration. The following criteria were used:

- a) if CVE is applicable to the relevant third-party library or simply contains a search string,
- b) if vulnerability is applicable, if it is applicable to the version used in the TOE (i.e. check if it is already patched in the version used),
- c) if it is clearly mitigated in the obvious manner (e.g. exploit requires shell access that is not offered by the TOE).

A pared-down list of remaining 64 matches then was sent to the vendor for further analysis. The vendor provided technical analysis and responded to the lab with additional details allowing to make final applicability determination.

The following search terms were utilized: SonicWall SMA, Linux kernel, intel Xeon E3 v6, intel Core i5-7500, sonicwallos, TCP/IP, acpi, acpi-support-base, openssl, openjdk, acpid, adduser, at, base-files, base-passwd, busybox, bzip2, coreutils, cpio, cron, dash, debianutils, diffutils, dpkg, e2fslibs:amd64, e2fsprogs, ethtool, file, findutils, gcc-4.9-base:amd64, grep, gzip, hostname, ifupdown, inetutils-ping, init, init-system-helpers, initscripts, insserv, iproute2, kmod, less, libacl1:amd64, libapt-pkg4.12:amd64, libattr1:amd64, libaudit-common, libaudit1:amd64, libblkid1:amd64, libbz2-1.0:amd64, libc-bin, libc6:amd64, libcomerr2:amd64, libdb5.3:amd64, libdebconfclient0:amd64, libgcc1:amd64, libgdbm3:amd64, libkmod2:amd64, liblzma2, liblzma5:amd64, libmagic1:amd64, libmount1:amd64, libpam-modules:amd64, libpam-modules-bin, libpam-runtime, libpam0g:amd64, libpcre3:amd64, libperl4-corelibs-pe, libpng12-0:amd64, libpopt0:amd64, libprocps3:amd64, libselinux1:amd64, libsemanage-common, libsemanage1:amd64, libsepol1:amd64, libslang2:amd64, libsmartcols1:amd64, libss2:amd64, libstdc++6:amd64, libtinfo5:amd64, libusb-0.1-4:amd64, libustr-1.0-1:amd64, libuuid1:amd64, locales, login, logrotate, lsb-base, lsof, makedev, mawk, mksh, module-init-tools, mount, multiarch-support, ncurses-term, net-tools, netbase, passwd, patch, pdksh, perl, perl-base, perl-modules, procps, psmisc, readline-common, rsync, sed, startpar, strace, sysv-rc, sysvinit, sysvinit-core, sysvinit-utils, tar, telnet, time, traceroute, tzdata, util-linux, vim-common, vim-tiny, xz-utils, 3ware Storage (RAID), Erlang OTP, Flask, Flask-RESTful, Jinja2, LZ4, MarkupSafe, PyMySQL, Werkzeug, aniso8601, apache-ant, apache-commons-beanutils, apache-commons-chain, apache-commons-codec, apache-commons-collections, apache-commons-dbc, apache-commons-digester, apache-commons-discovery, apache-commons-el, apache-commons-fileupload, apache-commons-httpclient, apache-commons-httpcomponents, apache-commons-io, apache-commons-lang, apache-commons-logging, apache-commons-net, apache-commons-pool, apache-commons-validator, apache-log4j, apache-maven, apache-struts1, apache-taglib, apache-velocity, apache-xalan-j, apache-xerces, apache-xmlrpc, apr, apr-util, bash, busybox, cJSON, cabextract, click, crash, curl, cyrus-sasl, dhcpcd, dialog, dmidecode, e2fsprogs, eventlog, gdb, geoView, ghostscript, glib, googletest, grub, gsoap, haveged, heimdal, hibernate-validator, httpd, icu, image4j, iniparser, iptables, itsdangerous, jackson, javamail, jersey, jetty, jfreechart, json-cpp, jsoup, junit, kexec-tools, legacy-spidermonkey, libcups, libdnet, libesmt, libevent, libgd, libmaxminddb, libmnl, libnftnl, libntlm, libpcap, libssh2, libxml2, libxslt, log4shib, mDNSResponder, mariadb-connector-c, mariadb-java-client, ncurses, net-snmp, nhttp2, nginx, node.js, ntp, open-vm-tools, opensv, openldap, opensaml, openssl, pciutils, pcre, pycrypto, python-dateutil, python-magic, pytz, readline, requests, rng-tools, samba, semver, six, slf4j, spidermonkey, stunnel, syslog-ng, tcpdump, uWSGI, valgrind, virtualbox, vlan, xerces-c, xmlsecurity, xmltooling, xz, zlib.

The evaluator searched the following public vulnerability repositories:

- The National Vulnerability Database at <https://nvd.nist.gov/vuln>
- The CVE Details website at <https://www.cvedetails.com/vulnerability-search.php>

SonicWALL SMA V12.4

Assurance Activity Report

Both sites were checked using the search terms listed above, package name and version provided in the list, as in many instances, one website would yield one or more results while the other provided no results, and vice versa. In many instances, several hundred potential vulnerabilities had to be checked for applicability. In every instance where each website generated hits, the results were cross checked for duplicate entries.

This list was cross-checked for completeness with the results of automated scanners (e.g., Nmap, Nessus) and TOE's self-reporting capabilities. Based on the module and component list, the evaluator conducted a vulnerability search using publicly available sources to identify potential vulnerabilities. The identified potential vulnerabilities were communicated to the vendor for further analysis and mitigation.

For each identified potential vulnerability, the evaluator recommended to patch, eliminating the vulnerability entirely. Failing that, the vendor had to provide a rationale explaining if vulnerability is applicable to the TOE and whether it is feasible to exploit it in the evaluated configuration. All identified vulnerabilities were either patched, deemed not applicable, or deemed infeasible. No attack potential analysis was necessary, and no residual vulnerabilities are known to be present in the product.

The evaluator confirmed through scanning and vendor's affirmation that the evaluated product includes fixes to resolve most of the vulnerabilities identified during the search and the newly identified vulnerabilities will be addressed according to a public CVE policy <https://psirt.global.sonicwall.com/vuln-policy> leaving no unresolved residual vulnerabilities.

- (2) The evaluator examined the TOE architecture and noted that it utilizes a database to store user data. The evaluator theorized that it is possible the TOE would be vulnerable to SQL injection attacks through the main web-based administrative interface. The evaluator devised a set of penetration tests targeting SQL injection to the specific version of database. The evaluator was unsuccessful in carrying out SQL injections as documented in the ETR AVA_VAN.1.

The TOE is not vulnerable to the ROBOT Padding Oracle attack, as the TOE's TLS server does not use the algorithm RSA as a TLS key exchange algorithm.