ASSURANCE ACTIVITY REPORT
FORTIGATE NGFW APPLIANCES RUNNING FORTIOS 5.4

Reference	EFS-T045-AAR	Status	Released
Version	1.1	Release Date	03 August 2018
Author	Will Gibbs	Customer	Fortinet, Inc.
Approved By	8/3/2018 X Will Gibbs 17025 Signatory Signed by: will.gibbs@baesystems.com		

BAE SYSTEMS
INSPIRED WORK

Table of Contents

1 INTRODUCTION

1.1 Overview

This report documents the Common Criteria FWcPP (with IPSEP and VPNEP) evaluation of FortiGate NGFW appliances running FortiOS 5.4 (FortiOS 5.4). A description of the assurance activities performed by the evaluators and their associated results are provided.

1.2 Evaluation details

Developer	Fortinet, Inc.
Sponsor	Fortinet, Inc.
Evaluator	BAE Systems Lab - AISEF
Scheme	AISEP
Task ID	EFS-T045

1.3 AAR configuration control identifiers

AAR Identifier	EFS-T045-AAR
AAR Title	Assurance Activity Report - FortiGate NGFW appliances running FortiOS 5.4
AAR Version/Date	1.1, 03 August 2018

1.4 ST configuration control identifiers

ST Title	FortiGate NGFW appliances running FortiOS 5.4
ST Version/Date	Version 1.1, 03-Aug-2018

1.5 TOE Configuration

TOE Name	FortiGate NGFW appliances running FortiOS 5.4 (FortiOS 5.4)
TOE Version	5.4

1.6 References

1.6.1 Requirements

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1, Revision 4

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 3.1, Revision 4

[3] Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, version 3.1 Revision 4

[4] Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1, Revision 4

[5] Collaborative Protection Profile for Stateful Traffic Filter Firewalls, Version 1.0, 27 Feb,2015

[6] Extended Package (EP) for Intrusion Prevention Systems (IPS), Version 2.1, 28 January 2016

[7] Network Device collaborative Protection Profile (NDcPP) Extended Package VPN Gateway, Version 2.1, 08 March 2017

1.6.2 Evaluation Evidence

[8] Security Target - FortiGate NGFW appliances running FortiOS 5.4, Version 1.0, 30 November 2017

[9] FIPS 140-2 and Common Criteria Compliant Operation for FortiOS™ 5.4, 12 October 2017

[10] FortiOS 5.4.3 FortiOS Log Reference December 21 2016

[11] FortiOS™ Handbook 5.4.3, 10 January 2017

1.6.3 Other References

[12] FIPS PUB 186-4 Digital Signature Standard, July 2013

1.7 Copyright statement

2 FWCPP - SFR ASSURANCE ACTIVITIES

This section of the AAR defines each of the SFRs specified in the ST (Ref. [8]), their corresponding assurance activities and the evaluator's findings in each case.

2.1 Security Audit (FAU)

2.1.1 FAU_GEN.1 Audit data generation

TSS	N/A

N/A

Guidance	The evaluator shall check the guidance documentation and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the cPP is described and that the description of the fields contains the information required in FAU_GEN1.2, and the additional information specified in the table of audit events. The evaluator shall also make a determination of the administrative actions that are relevant in the context of the cPP. The evaluator shall examine the guidance documentation and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are security relevant with respect to the cPP. The evaluator may perform this activity as part of the activities associated with ensuring that the corresponding guidance documentation satisfies the requirements related to it.

The FortiOS Log Reference (Ref. [10]) provides detailed information on all logs generated by the TOE and covers each field mandated by the cPP as well as providing additional information on many additional fields.

Additionally, the 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref.[9]) provides instructions that need to be followed in order for the TOE to meet the logging requirements of the cPP.

Testing	The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the table of audit events and administrative actions listed above. This should include all instances of an event: for instance, if there are several different I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. Logging of all activities related to trusted update should be tested in detail and with utmost diligence. 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.

Throughout the testing performed, the evaluators examined the TOE to determine whether audit log entries for all auditable events were generated. The evaluators confirmed that all auditable events are generated.

The evaluators confirmed that audit entries were generated for the firewall rules for permit, deny and log.

The evaluators confirmed that, in the scenario when the TOE is subjected to more traffic than the interfaces are able to handle, the TOE automatically drops all received packets until the overwhelming traffic ceases and audits these events appropriately.

2.1.2 FAU_STG_EXT.1 Protected audit event storage

TSS	The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided. The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access. If the TOE complies with FAU_STG_EXT.2 the evaluator shall verify that the numbers provided by the TOE according to the selection for FAU_STG_EXT.2 are correct when performing the tests for FAU_STG_EXT.1.3. 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.

Logs are written to the FortiGate unit hard disk if the unit contains one. Models that do not contain a hard disk log to system memory. The amount of audit data that can be stored is dependent on the capacity of the device.

Local log files can only be deleted via the CLI by an authorised administrator. No editing of log data is permitted.

When the TOE is configured to transmit log data to an external FortiAnalyzer platform (via TLS), log data is cached prior to transmission. As such, no modification or deletion of the log data is possible.

If the local storage for audit logs is filled, the oldest stored logs will be deleted in a First-In-First-Out (FIFO) order to allow for the storage of new events.

Guidance	The evaluator shall also examine the guidance documentation to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server. The evaluator shall also examine the guidance documentation to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server. The evaluator shall also ensure that the guidance documentation describes all possible configuration options for FAU_STG_EXT.1.3 and the resulting behaviour of the TOE for each possible configuration. The description of possible configuration options and resulting behaviour shall correspond to those described in the TSS.

The 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) configuration necessary to connect to the supported external logging device and explains how logs are cached locally on the Fortinet device before been offloaded to the external logging server.

Additional information on the configuration of the external Fortinet logging device is available in chapter 18 of FortiOS Handbook (Ref. [11]).

Testing	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: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing. The evaluator shall 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: a) 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). b) 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) c) The TOE behaves as specified (for the option 'other action' in FAU_STG_EXT.1.3).

Evaluators configured the TOE and the FortiAnalyzer in line with the guidance provided. Evaluators performed a number of events to generate audit log data and confirmed that the audit traffic was encrypted and unable to be viewed in the clear.

The evaluators tested that upon filling up the existing log storage, newly created logs began overwriting the existing records starting with the oldest records.

2.2 Cryptographic Support (FCS)

2.2.1 FCS_CKM.1 Cryptographic Key Generation

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

Table 14 within the TSS indicates that the TOE generates asymmetric cryptographic keys in accordance with FIPS PUB 186-4

Guidance	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 uses defined in this PP.

The "Managing X.509 certificates" section of chapter 4 of FortiOS Handbook (Ref. [11]) describes key generation and establishment configuration of the TOE.

Testing	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.

The key generation functionality implemented within the TOE has been validated via the CAVP. The applicable algorithm certificate numbers are listed within the ST (Ref. [8]).

2.2.2 FCS_CKM.2 Cryptographic Key Establishment

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

Table 15 within the TSS indicates that the TOE generates asymmetric cryptographic keys in accordance with SP 800-56B

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

The "Managing X.509 certificates" section of chapter 4 of FortiOS Handbook (Ref. [11]) describes key generation and establishment configuration of the TOE.

Testing	The evaluator shall verify the implementation of the key establishment schemes of the supported by the TOE.

The key generation functionality implemented within the TOE has been validated via the CAVP. The applicable algorithm certificate numbers are listed within the ST (Ref. [8]).

2.2.3 FCS_CKM.4 Cryptographic Key Destruction

TSS	The evaluator shall check to ensure the TSS lists each type of plaintext key material and its origin and storage location. The evaluator shall verify that the TSS describes when each type of key material is cleared (for example, on system power off, on wipe function, on disconnection of trusted channels, when no longer needed by the trusted channel per the protocol, etc.). The evaluator shall also verify that, for each type of key, the type of clearing procedure that is performed (cryptographic erase, overwrite with zeros, overwrite with random pattern, or block erase) is listed. If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the clearing procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are cleared by overwriting once with zeros, while secret keys stored on the internal persistent storage device are cleared by overwriting three times with a random pattern that is changed before each write").

The TOE stores keys in memory (RAM or Flash). The key destruction methods implemented by the TOE meet the requirements of FIPS PUB 140-2 Level 1.

The TOE provides the following zeroization methods for cryptographic keys and other material:

- Volatile memory (SDRAM): The TOE performs a single direct overwrite consisting of zeroes, followed by a read-verify. If the read-verification of the overwritten data fails, the process repeats.
- Non-volatile flash memory (Flash RAM): The TOE performs a single, direct overwrite consisting of zeroes, which is followed by a followed by a read-verify. If the read-verification fails, the process repeats.

Guidance	N/A

N/A

Testing	N/A

N/A

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

TSS	N/A

Table 16 within the TSS indicates that the TOE perform encryption/decryption utilising AES in CBC and GCM modes, with cryptographic key sizes 128 and 256 bits in accordance with:

- AES as specified in ISO 18033-3;
- CBC as specified in ISO 10116; and
- GCM as specified in ISO 19772.

Guidance	N/A

N/A

Testing	The evaluator shall use tests appropriate to the modes selected in the above requirement from "The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)", "The XTS-AES Validation System (XTSVS)", The CMAC Validation System (CMACVS)", "The Counter with Cipher Block ChainingMessage Authentication Code (CCM) Validation System (CCMVS)", and "The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS)" (these documents are available from http://csrc.nist.gov/groups/STM/cavp/index.html) as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

The cryptographic functionality implemented within the TOE has been validated via the CAVP. The applicable algorithm certificate numbers are listed within the ST (Ref. [8]).

2.2.5 FCS_COP.1(2) Cryptographic Operation (Signature Generation and Verification)

TSS	N/A

Table 16 within the TSS indicates that the TOE provides cryptographic signature services (generation and verification) using the RSA algorithm (with a 2048-bit modulus) in accordance with FIPS PUB 186-4.

Guidance	N/A

N/A

Testing	The evaluator shall use the signature generation and signature verification portions of "The Digital Signature Algorithm Validation System" (DSAVS or DSA2VS), "The Elliptic Curve Digital Signature Algorithm Validation System" (ECDSAVS or ECDSA2VS), and "The RSA Validation System" (RSAVS) as a guide in testing the requirement above. The Validation System used shall comply with the conformance standard identified in the ST (i.e., FIPS PUB 186-2 or FIPS PUB 186-3). This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

The cryptographic functionality implemented within the TOE has been validated via the CAVP. The applicable algorithm certificate numbers are listed within the ST (Ref. [8]).

2.2.6 FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)

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

Table 16 within the TSS indicates that the TOE performs cryptographic hashing utilising SHA-1, SHA-256, SHA-384 and SHA-512 in accordance with ISO/IEC 10118-3:2004.

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

Chapter 16 of the FortiOS handbook contains the configuration information for configuring the hashing algorithm used for IPsec connections.

Testing	The TSF hashing functions can be implemented in one of two modes. The first mode is the byte oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit oriented vs. the byte oriented test macs. The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP.

The cryptographic functionality implemented within the TOE has been validated via the CAVP. The applicable algorithm certificate numbers are listed within the ST (Ref. [8]).

2.2.7 FCS_COP.1(4) Cryptographic Operation (Keyed Hash Algorithm)

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

Table 16 within the TSS indicates that the TOE performs hashed message authentication utilising HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512, with cryptographic key sizes of 160, 256, 384 and 512 bits and message digest sizes of 160, 256, 384 and 512 bits that meet the requirements of ISO/IEC 9797-2:2011, Section 7 ("MAC Algorithm 2").

Guidance	N/A

N/A

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

The cryptographic functionality implemented within the TOE has been validated via the CAVP. The applicable algorithm certificate numbers are listed within the ST (Ref. [8]).

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

TSS	N/A

N/A

Guidance	N/A

N/A

Testing	The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the guidance documentation contains appropriate instructions for configuring the RNG functionality. If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. "generate one block of random bits" means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP800-90A). If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call. The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator. **Entropy input:** the length of the entropy input value must equal the seed length. **Nonce:** If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length. **Personalization string:** The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied. **Additional input**: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.

The cryptographic functionality implemented within the TOE has been validated via the CAVP. The applicable algorithm certificate numbers are listed within the ST (Ref. [8]).

2.2.9 FCS_HTTPS_EXT.1 HTTPS Protocol

TSS	N/A

N/A

Guidance	N/A

N/A

Testing	The evaluator shall perform the following tests: • The evaluator shall attempt to establish an HTTPS connection with a web server, observe the traffic with a packet analyzer, and verify that the connection succeeds and that the traffic is identified as TLS or HTTPS. Other tests are performed in conjunction with the TLS evaluation activities. Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1, and the evaluator shall perform the following test: • The evaluator shall demonstrate that using a certificate without a valid certification path results in an application notification. • Using the administrative guidance, the evaluator shall then load a valid certificate and certification path, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the selection listed in the ST occurs.

The evaluators attempted to establish a connection to the TOE using HTTPS and observed via packet analyser that the connection succeeded and was using the TLS/HTTPS protocol. The evaluator then attempted the same connection using a certificate for authentication and also observed the connection succeeded.

Evaluators then attempted to establish an HTTPS connection with an invalid certificate and confirmed that the attempt failed.

2.2.10 FCS_SSHS_EXT.1 SSH Server

2.2.10.1 FCS_SSHS_EXT.1.2

TSS	The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSHS_EXT.1.5, and ensure that password-based authentication methods are also allowed.

Section 8.5 of the ST indicates that the TOE implements SSH in compliance with RFCs 4251 through 4254, 5647, 5656, 6187 and 6668.

The TOE supports password-based or public key (SSH-RSA) authentication.

Guidance	N/A

N/A

Testing	The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection. Any configuration activities required to support this test shall be performed according to instructions in the guidance documentation. The evaluator shall choose one public key algorithm supported by the TOE. The evaluator shall generate a new key pair for that algorithm without configuring the TOE to recognize the public key for authentication. The evaluator shall use an SSH client to attempt to connect to the TOE with the new key pair and demonstrate that authentication fails. Using the guidance documentation, the evaluator shall configure the TOE to accept password-based authentication, and demonstrate that a user can be successfully authenticated to the TOE over SSH using a password as an authenticator. The evaluator shall use an SSH client, enter an incorrect password to attempt to authenticate to the TOE, and demonstrate that the authentication fails.

The evaluators attempted to connect using each allowed public key algorithm and confirmed that each connection attempt succeeded. The evaluators attempted to authenticate to the TOE using a public key not associated with any entity – this connection attempt was rejected.

The evaluators attempted to establish an SSH connection after configuring the TOE to permit password-based authentication and confirmed the connection succeeded when correct credentials were provided. The test was repeated using an incorrect password and it was confirmed that the TOE rejected the connection attempt.

2.2.10.2 FCS_SSHS_EXT.1.3

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

The TOE examines the size of each received SSH packet - if a packet is greater than 32768 bytes, it is automatically dropped.

Guidance	N/A

N/A

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

The evaluators connected to the TOE using a script that initiated the SSH handshake and transmitted a packet larger than 32768 bytes. Evaluators observed that the connection failed and confirmed that the connection failure was recorded in the traffic log.

2.2.10.3 FCS_SSHS_EXT.1.4

TSS	The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.

The TOE utilises AES-CBC-128 and AES-CBC-256 for SSH encryption.

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

The 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) contains instructions for placing the device into FIPS/CC mode, which restricts the TOE to using only the algorithms specified in this requirement.

Testing	The evaluator shall establish a SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test. The evaluator shall configure an SSH client to only allow the 3descbc encryption algorithm and no other encryption algorithms. The evaluator shall attempt to establish an SSH connection from the SSH client to the TOE and observe that the connection is rejected.

The evaluators connected to the TOE using each specified encryption algorithm and confirmed that each attempt succeeded.

The evaluators attempted to connect to the TOE specifying 3DES-CBC as the encryption algorithm and confirmed that the connection attempt was rejected.

2.2.10.4 FCS_SSHS_EXT.1.5

TSS	The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the public key algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the public key algorithms specified are identical to those listed for this component.

The TOE supports public key (SSH-RSA) authentication.

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

The 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) contains instructions for placing the device into FIPS/CC mode, which restricts the TOE to using only the algorithms specified in this requirement.

Testing	The evaluator shall establish a SSH connection using each of the public key algorithms specified by the requirement to authenticate the TOE to an SSH client. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test. The evaluator shall configure an SSH client to only allow the ssh-dsa public key algorithm and no other public key algorithms. The evaluator shall attempt to establish an SSH connection from the SSH client to the TOE and observe that the connection is rejected.

The evaluators connected to the TOE using each specified public key algorithm and confirmed that each attempt succeeded.

The evaluators attempted to connect to the TOE specifying SSH-DSA as the public key algorithm and confirmed that the connection attempt was rejected.

2.2.10.5 FCS_SSHS_EXT.1.6

TSS	The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component.

The TOE provides data integrity for SSH connections via HMAC-SHA1, HMAC-SHA2-256 and HMAC-SHA2-512.

Guidance	The evaluator shall also check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the "none" MAC algorithm is not allowed).

The 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) contains instructions for placing the device into FIPS/CC mode, which restricts the TOE to using only the algorithms specified in this requirement.

Testing	The evaluator shall establish a SSH connection using each of the integrity algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test. The evaluator shall configure an SSH client to only allow the "none" MAC algorithm. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails. The evaluator shall configure an SSH client to only allow the hmacmd5 MAC algorithm. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.

The evaluators connected to the TOE using each specified integrity algorithm and confirmed that each attempt succeeded.

The evaluators attempted to connect to the TOE specifying either HMAC-MD5 or 'none' as the chosen integrity algorithm and confirmed that the connection attempt was rejected.

2.2.10.6 FCS_SSHS_EXT.1.7

TSS	The evaluator shall check the TSS to ensure that it lists the supported key exchange algorithms, and that that list corresponds to the list in this component.

The TOE supports Diffie-Hellman Group 14 SHA-1 (diffie-hellman-group14-sha1) for SSH key exchanges.

Guidance	The evaluator shall also check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed key exchange algorithms are used in SSH connections with the TOE.

The 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) contains instructions for placing the device into FIPS/CC mode, which restricts the TOE to using only the algorithms specified in this requirement.

Testing	For each allowed key exchange method, the evaluator shall configure an SSH client to only allow that method for key exchange, attempt to connect from the client to the TOE, and observe that the attempt succeeds. The evaluator shall configure an SSH client to only allow the diffiehellman-group1-sha1 key exchange. The evaluator shall attempt to connect from the SSH client to the TOE and observe that the attempt fails.

The evaluators connected to the TOE using each specified key exchange method and confirmed that each attempt succeeded.

The evaluators attempted to connect to the TOE specifying diffiehellman-group14-sha1 as the chosen key exchange method and confirmed that the connection attempt was rejected.

2.2.10.7 FCS_SSHS_EXT.1.8

TSS	N/A

Guidance	N/A

Testing	The evaluator shall configure the TOE to create a log entry when a rekey occurs. The evaluator shall connect to the TOE with an SSH client and cause 2^28 packets to be transmitted from the client to the TOE, and subsequently review the audit log to ensure that a rekey occurred.

The evaluators established an SSH connection to the TOE and forced the connection to stay active long enough to generate 2^28 packets. The evaluators confirmed that a log entry was generated by the TOE after exceeding 2^28 packets and rekeying occurred.

2.2.11 FCS_TLSC_EXT.2 Extended: TLS Client Protocol with authentication

2.2.11.1 FCS_TLSC_EXT.2.1

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

Per Section 8.4 of the ST, the TOE generates RSA keys of 2048-bit size, uses NIST curve secp256r1 and generates Diffie-Hellman parameters of 2048-bit size for use in key agreement/key exchange messages.

The TOE supports the Supported Elliptic Curves Extension by default.

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

The TOE utilises TLS for connections to an external logging device. The instructions required to configure this connection are provided in the "Log-specific Settings" section of the 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]).

Testing	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 in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES). 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. The evaluator shall send a server certificate in the TLS connection that the does not match the server-selected ciphersuite (for example, send a 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. The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection. Test 2 in FCS_TLSS_EXT.1.1 or FCS_TLSS_EXT.2.1 can be used as a substitute for this test. The evaluator perform 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 (for example 1.3 represented by the two bytes 03 04) and verify that the client rejects the connection. b) 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. c) 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. d) Modify the signature block in the Server's Key Exchange handshake message, and verify that the client rejects the connection after receiving the Server Key Exchange message. e) Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data. f) Send a garbled message from the Server after the Server has issued the ChangeCipherSpec message and verify that the client denies the connection.

The evaluators attempted to establish an SSH connection between the TOE and the logging device using each of the ciphersuites specified by the requirement. The evaluators confirmed that, for each algorithm, the connection was successfully established.

The evaluator attempted connecting to the external logging device from the TOE using the 'non-standard' certificate variations and traffic modifications specified in this requirement. All variations of traffic modification and certificate modification failed.

2.2.11.2 FCS_TLSC_EXT.2.2

TSS	The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the administrator/applicationconfigured reference identifier, including which types of reference identifiers are supported (e.g Common Name, DNS Name, URI Name, Service Name, or other application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported. The evaluator shall ensure that this description identifies whether and the manner in which certificate pinning is supported or used by the TOE.

Section 8.4 of the ST states the TOE establishes reference identifiers using the following:

- Host IP address; and/or
- Fully qualified domain name (FQDN).

Wildcards are supported. Certificate pinning is not supported/used.

Guidance	The evaluator shall verify that the AGD guidance includes instructions for setting the reference identifier to be used for the purposes of certificate validation in TLS.

The TOE utilises TLS for connections to an external logging device. The instructions required to configure this connection are provided in the "Log-specific Settings" section of the 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]). This includes a note that the reference identifier set as part of the "set server" cli command can be either a FQDN or an ip address.

Testing	The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection: a) The evaluator shall present a server certificate that does not contain an identifier in either the Subject Alternative Name (SAN) or Common Name (CN) that matches the reference identifier. The evaluator shall verify that the connection fails. b) 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. c) The evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contains the SAN extension. The evaluator shall verify that the connection succeeds. d) 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. e) The evaluator shall perform the following wildcard tests with each supported type of reference identifier: a. 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. b. 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. 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.come) and verify that the connection fails. f) [conditional] If URI or Service name reference identifiers are supported, the evaluator shall configure the DNS name and the service identifier. The evaluator shall present a server certificate containing the correct DNS name and service identifier in the URIName or SRVName fields of the SAN and verify that the connection succeeds. The evaluator shall repeat this test with the wrong service identifier (but correct DNS name) and verify that the connection fails. g) [conditional] If pinned certificates are supported the evaluator shall present a certificate that does

The evaluators attempted to connect to the external logging device from the TOE while making each of the above modifications to the reference identifier. Connection attempts using each of the invalid variations of the reference identifier failed and were logged, whilst attempts using the valid reference identifiers succeeded.

2.2.11.3 FCS_TLSC_EXT.2.3

TSS	N/A

N/A

Guidance	N/A

N/A

Testing	The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. Using the administrative guidance, the evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. . If the certificate is validated and a trusted channel is established, the test passes. The evaluator then shall delete one of the certificates, and show that the certificate is not validated and the trusted channel is not established.

The evaluators attempted a connection from the TOE to the external logging device using a valid certificate, but without a valid certification path being in place. The evaluators confirmed that this connection attempted failed.

The evaluators re-configured the TOE to establish a valid certification path and confirmed that the connection was established.

Further modifications to the certificate path (i.e. removing one or more certificates) caused the connection attempts to fail.

2.2.11.4 FCS_TLSC_EXT.2.4

TSS	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.

Per Section 8.4 of the ST, the TOE supports the Supported Elliptic Curves Extension by default.

Guidance	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.

The TOE utilises TLS for connections to an external logging device. The instructions required to configure this connection are provided in the "Log-specific Settings" section of the 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]).

Testing	The evaluator shall configure the server to perform an ECDHE key exchange in the TLS connection using a non-supported curve (for example P-192) and shall verify that the TOE disconnects after receiving the server's Key Exchange handshake message.

The evaluators attempted to connect using a non-supported curve for ECDHE key exchange and confirmed the connection failed upon receipt of the server Key Exchange handshake message.

2.2.11.5 FCS_TLSC_EXT.2.5

TSS	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.

Per the TSS, the TOE uses X509 certificates in accordance with RFC 5280 for TLS mutual authentication.

Guidance	The evaluator shall verify that the AGD guidance required per FIA_X509_EXT.2.1 includes instructions for configuring the client-side certificates for TLS mutual authentication.

The TOE utilises TLS for connections to an external logging device. The instructions required to configure this connection are provided in the "Log-specific Settings" section of the 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]).

Testing	The evaluator shall perform the following modification to the traffic: • Configure the server to require mutual authentication and then modify a byte in a CA field in the Server's Certificate Request handshake message. The modified CA field must not be the CA used to sign the client's certificate. The evaluator shall verify the connection fails.

The evaluator attempted to connect to the logging server after modifying a byte in the CA field of the Server Certificate Request handshake message and observed that the connection failed.

2.2.12 FCS_TLSS_EXT.1 Extended: TLS Server Protocol

2.2.12.1 FCS_TLSS_EXT.1.1

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

Per Section 8.4 of the ST, the TOE implements TLS in accordance with RFCs 4346 (TLS v1.1) and 5246 (TLS v1.2).

The TOE supports the following cipher suites for TLS connections:

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268;
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268;
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246;
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246;
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268;
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268;
- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246;
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246;
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492;
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492;
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289; and
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289.

Guidance	The evaluator shall also 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).

The 'FIPS and CC Configuration Guide' (Ref. [9]) contains the steps needed to place the TOE into FIPS-CC mode. This prevents the use of any invalid TLS ciphersuites.

Testing	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 in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES). 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. The evaluator shall use a client to send a key exchange message in the TLS connection that the does not match the server-selected ciphersuite (for example, send an ECDHE key exchange while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite or send a RSA key exchange while using one of the ECDSA ciphersuites.) The evaluator shall verify that the TOE disconnects after the receiving the key exchange message. The evaluator shall perform the following modifications to the traffic: a) Modify at a byte in the client's nonce in the Client Hello handshake message, and verify that the server rejects the client's Certificate Verify handshake message (if using mutual authentication) or that the server denies the client's Finished handshake message. b) Modify the signature block in the Client's Key Exchange handshake message, and verify that the server rejects the client's Certificate Verify handshake message (if using mutual authentication) or that the server denies the client's Finished handshake message. c) Modify a byte in the Client Finished handshake message, and verify that the server rejects the connection and does not send any application data. d) After generating a fatal alert by sending a Finished message from the client before the client sends a ChangeCipherSpec message, send a Client Hello with the session identifier from the previous test, and verify that the server denies the connection. e) Send a garbled message from the client after the client has issued the ChangeCipherSpec message and verify that the Server denies the connection.

The evaluator established a TLS connection using each of the ciphersuites specified in the requirement. This connection was confirmed to be successful.

The evaluator attempted connecting to the TOE using all other variations of certificate and traffic modification specified above. All of these attempts were confirmed to fail.

2.2.12.2 FCS_TLSS_EXT.1.2

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

Per Section 8.4 of the ST, the TOE will deny connections where the requested protocol is SSL 1.0 through 3.0 and TLS 1.0.

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

The 'FIPS and CC Configuration Guide' (Ref.[9]) contains the steps needed to place the TOE into FIPS-CC mode. When in this mode, the TOE automatically rejects connection attempts SSL 1.0 through 3.0 and TLS 1.0.

Testing	The evaluator shall send a Client Hello requesting a connection with version SSL 1.0 and verify that the server denies the connection. The evaluator shall repeat this test with SSL 2.0, SSL 3.0, TLS 1.0, and any selected TLS versions.

The evaluators attempted to connect to the TOE using the invalid SSL and TLS versions specified in the requirement and confirmed that, in each instance, the connection was rejected.

2.2.12.3 FCS_TLSS_EXT.1.3

TSS	The evaluator shall verify that the TSS describes the key agreement parameters of the server key exchange message.

Per Section 8.4 of the ST, the TOE generates RSA keys of 2048-bit size, uses NIST curves secp256r1 and secp384r1 and generates Diffie-Hellman parameters of 2048-bit size for use in key agreement/key exchange messages

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

The 'FIPS and CC Configuration Guide' (Ref. [9]) contains the steps needed to place the TOE into FIPS-CC mode. This prevents the use of any invalid key agreement parameters.

Testing	The evaluator shall attempt a connection using an ECDHE ciphersuite and a configured curve and, using a packet analyzer, verify that the key agreement parameters in the Key Exchange message are the ones configured (determining that the size matches the expected size for the configured curve is sufficient.). The evaluator shall repeat this test for each supported NIST Elliptic Curve and each supported Diffie-Hellman key size.

The evaluator attempted to connect to the TOE using each supported ECDHE ciphersuite and curve and confirmed that the connection was denied.

2.3 User Data Protection (FDP)

2.3.1 FDP_RIP.2 Full Residual Information Protection

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

Section 8.8 of the ST states that the TOE ensures that no information from previously processed information flows is transferred to subsequent information flows. This applies both to information that is input to the TOE from an external source and to information (e.g., padding bits) that might be added by the TOE during processing of the information from the external source.

The removal of any previous residual information is achieved through the zeroization of data when the memory structure is initially created (to clear any residual/prior information) and strict bounds checking on the data prior to it being assigned in memory.

Guidance	N/A

N/A

Testing	N/A

N/A

2.4 Identification and Authentication (FIA)

2.4.1 FIA_PMG_EXT.1 Password Management

TSS	N/A

N/A

Guidance	The evaluator shall examine the guidance documentation to determine that it provides guidance to security administrators on the composition of strong passwords, and that it provides instructions on setting the minimum password length.

The 'FIPS and CC Configuration Guide' (Ref. [9]) describes how enabling FIPS-CC mode changes the password requirements of the administrator account.

The 'Password Policy' section of the FortiOS Handbook (Ref. [11]) provides information on the configuration of password settings and gives recommendations on how passwords should be composed.

Testing	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, rule characteristics, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.

The evaluators developed a list of passwords, a subset of which met the password requirements and a subset that did not. The evaluators tested each of these passwords (which included each permitted/non-permitted character in the standard ASCII range) and determined that they were accepted/rejected by the TOE as applicable.

2.4.2 FIA_UIA_EXT.1 User Identification and Authentication

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

Per Section 8.9 of the ST, the administrator login process is as follows:

- An entity provides a username/password or username/key to the TOE.
- If the username and password/key provided are valid and correct, the TOE ends the logon process and grants the administrator access to TOE functionality (a successful logon).
- If the provided credentials are invalid, the TOE presents an error message and no access to the TOE is provided (unsuccessful login).

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

Chapter 4 (Authentication) of the FortiOS Handbook (Ref.[11]) provides information on configuring the authentication methods provided by the TOE. Chapter 2 (Getting Started) provides guidance on how to log in to and administer the TOE.

Testing	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: 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. 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. 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.

The evaluators configured the password-based and public-key based authentication mechanisms supported by the TOE (for both local

and remote authentication, where applicable).

The evaluators confirmed that successfully completing the authentication process provides access to TOE functions, failed authentication attempts are met with an error and no access to TOE functionality is provided.

No services are available to TOE users prior to authentication.

2.4.3 FIA_UAU.7 Protected Authentication Feedback

TSS	N/A

N/A

Guidance	N/A

N/A

Testing	The evaluator shall perform the following test for each method of local login allowed: • 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.

The evaluators authenticated locally to the TOE and confirmed that no feedback (except in the case of errors) is provided to the TOE during the authentication attempts.

2.4.4 FIA_X509_EXT.1 X.509 Certificate Validation

TSS	The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place. The evaluator ensures the TSS also provides a description of the certificate path validation algorithm.

Per Section 8.10 of the ST, the TOE utilises X509 certificates in accordance with RFC 5280. The TOE performs validation of certificates during the handshaking process for both HTTPS and IPsec connections.

The TOE validates certificates via Certificate Revocation List (CRL, RFC 5759). The validation is performed in the following steps:

- The remote client sends its certificate and associated key to the TOE.
- The TOE compares the received certificate and key against the certificate store (CA certificates, remote certificates, etc.) to determine that the certificate is authentic.
- The TOE then compares the certificate against any loaded CRLs.
- If the certificate is determined to be invalid or revoked, the certificate is rejected and the connection is not established
- If the certificate is determined to be valid, the connection process continues.
- If the certificate path does not terminate with a trusted CA, the validation will fail.

The TOE will reject CA certificates that lack the basicConstraints section or contain the section but whose CA flag is not set to TRUE

Guidance	N/A

N/A

Testing	The evaluator shall perform the following tests for FIA_X509_EXT.1.1: The evaluator shall demonstrate that validating a certificate 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 to be used in the function, and demonstrate that the function succeeds. The evaluator shall then delete one of the certificates, and show that the function fails. The evaluator shall demonstrate that validating an expired certificate results in the function failing. The evaluator shall test that the TOE can properly handle revoked certificates-–conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the TOE certificate and revocation of the TOE intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. 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. 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.) The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.) The evaluator shall modify any byte in the public key of the certificate and demonstrate that the cert The evaluator shall perform the following tests for FIA_X509_EXT.1.2. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. The evaluator shall create a chain of at least four certificates: the node certificate to be tested, two intermediate CAs, and the self-signed Root CA. a) The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate does not contain the basicConstraints extension. The validation of the certificate path fails. b) The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension set to FALSE. The validation of the certificate path fails. c) The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension set to TRUE. The validation of the

The evaluator attempted the above certificate and certificate path modifications. All variations of certificate and certificate path first failed.

2.4.5 FIA_X509_EXT.2 X.509 Certificate Authentication

TSS	The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates. The evaluator shall examine the TSS to confirm that it describes the behaviour of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the guidance documentation contains instructions on how this configuration action is performed.

Per Section 8.10 of the ST, the TOE utilises X509 certificates in accordance with RFC 5280.

The TOE performs validation of certificates during the handshaking process for both HTTPS and IPsec connections. The certificate to be used for a given connection is configured by the administrator as part of the applicable policy (IPsec, TLS, etc.).

If the TOE is unable to use a CRL for determining certificate validity, the TOE will not establish a connection (i.e. the TOE does not accept the presented certificate).

Guidance	N/A

N/A

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

The evaluator attempted to connect to the TOE (via HTTPS and/or IPsec) using the specified trusted channel, but configured the certificate chain in such a way that the TOE was unable to validate the provided client certificate. As per the choice in X509_EXT.2.2 the connection was refused when it was unable to validate the certificate.

2.4.6 FIA_X509_EXT.3 Extended: X509 Certificate Requests

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

Per Section 8.10 of the ST, the TOE will include the following information in generated CSRs:

- Certification Name;
- Subject Information;
- Organizational Unit;
- Organization;
- Locality (City);

- State/Province;
- Country/Region;
- E-mail;
- Key Type;
- Key Size;
- HSM; and
- Enrolment Method: File Based or Online SCEP.

The TOE validates certificates via Certificate Revocation List (CRL, RFC 5759). Certificate validation takes place during the handshake of HTTPS or IPsec connections.

Guidance	The evaluator shall check to ensure that the guidance documentation contains instructions on requesting certificates from a CA, including generation of a Certificate Request Message. 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 certificate request message.

The Certificate-based authentication section of the handbook (Ref.[11]) provides guidance on generating a certificate signing request.

Testing	The evaluator shall perform the following tests: a) The evaluator shall use the guidance documentation to cause the TOE to generate a certificate request message. The evaluator shall capture the generated message and ensure that it conforms to the format specified. The evaluator shall confirm that the certificate request provides the public key and other required information, including any necessary user-input information. b) The evaluator shall demonstrate that validating a certificate response message 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. The evaluator shall then delete one of the certificates, and show that the function fails.

The evaluators created a new certificate request using the TOE. Evaluators examined the CSR once downloaded to the Certificate Authority and confirmed it contained the following information fields:

- Certification Name;
- Subject Information;
- Organizational Unit;
- Organization;
- Locality (City);
- State/Province;
- Country/Region;
- E-mail;
- Key Type;
- Key Size;
- HSM; and
- Enrolment Method: File Based or Online SCEP.

The evaluators then attempted to validate a certificate response message without providing a valid path – this attempt resulted in a failure. Upon configuring the TOE with the certificates needed, the validation was confirmed to succeed. Removing these certificates then caused the validation to once again fail.

2.5 Security management (FMT)

2.5.1 FMT_MOF.1(1)/TrustedUpdate

TSS	N/A

N/A

Guidance	N/A

N/A

Testing	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). This test should fail. The evaluator shall try to perform the update with prior authentication as security administrator using a legitimate update image. This test should pass. This test case should be covered by the tests for FPT_TUD_EXT.1 already.

The TOE defines a single role, that of the Security Administrator. No access to TOE functionality, configuration or management can be performed without prior successful authentication.

Testing of the update mechanism implemented by the TOE is performed as part of the test activities for FPT_TUD_EXT.1.

2.5.2 FMT_MTD.1 Management of TSF Data

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

Per Section 8.11 of the ST, the TOE does not permit access to any functions (other than the warning/consent banner and authentication interface) prior to login. The TOE defines a single role, which is that of the Security Administrator.

The Security Administrator is able to perform the following management functions:

- Administer the TOE locally and remotely;
- Configure the access banner;
- Configure the session inactivity time before session termination or locking;
- Update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- Configure the cryptographic functionality;
- Modify, delete, generate and/or import cryptographic keys;
- Configure the IPsec functionality;
- Import X.509v3 certificates;
- Enable, disable, determine and modify the behavior of all the security functions of the TOE identified in this EP to the Administrator;

- Ability to configure firewall rules;
- Enable/disable signatures applied to sensor interfaces and determine the behavior of IPS functionality;
- Modify the parameters that define the network traffic to be collected and analyzed:
  - Source IP addresses (host address and network address)
  - Destination IP addresses (host address and network address)
  - Source port (TCP and UDP)
  - Destination port (TCP and UDP)
  - Protocol (IPv4 and IPv6)
  - ICMP type and code
- Update (import) signatures;
- Create custom signatures;
- Configure anomaly detection;
- Enable and disable actions to be taken when signature or anomaly matches are detected;
- Modify thresholds that trigger IPS reactions;
- Modify the duration of traffic blocking actions;
- Modify the known-good and known-bad lists (of IP addresses or address ranges); and
- Configure the known-good and known-bad lists to override signature-based IPS policies.

Guidance	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.

The 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) states that the only supported role in evaluated configuration is the administrator and as such they are the only ones able to configure and manipulate the TOE.

Testing	N/A

N/A

2.5.3 FMT_SMR.2 Restrictions on security roles

TSS	N/A

N/A

Guidance	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.

Chapter 4 – Authentication of the FortiOS handbook (Ref.[11]) provides information on configuring the authentication methods provided by the TOE. Chapter 2 ('Getting Started') provides guidance on how Administrators can log on to and administer the TOE.

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

During testing, the evaluators administered the TOE both locally and remotely and confirmed that both local and remote administration methods function as described in the guidance documentation.

2.6 Protection of the TSF (FPT)

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

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

Per Section 8.12 of the ST, the TOE prevents the reading of all pre-shared, symmetric and private keys stored within the TOE boundary. The TOE does not provide any interface for a user to do so.

Guidance	N/A

N/A

Testing	N/A

N/A

2.6.2 FPT_APW_EXT.1 Protection of Administrator Passwords

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

Per Section 8.12 of the ST, once a password is entered the TOE encrypts the password using AES-128 and writes the password to the configuration file, permanently obscuring the contents.

The configuration file (with the encrypted password hashes) is available through the local console and HTTPS GUI by viewing a full configuration or backup of the configuration.

Guidance	N/A

N/A

Testing	N/A

N/A

2.6.3 FPT_TST_EXT.1 TSF testing

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

Per Section 8.12 of the ST, the TOE performs the following self-tests upon initialisation:

- **CPU and Memory BIOS self-tests**

  CPU and memory are initialized by exercising a set of known answer tests and the BIOS is compared against a known checksum of the image. The memory is zeroized and then has a random pattern written and read from the memory.
- **Boot loader image verification**

  The boot loader will compare the image of the TOE to a known checksum of the image prior to booting.
- **Noise source tests**

  The noise source is started and pattern analysis is done on the output to ensure that the source is not stuck in a cryptographically weak state. These include both the repetition and adaptive proportion tests
- **FIPS 140-2 Known Answer Tests (KAT)**

  Comparison of a number of cryptographic functions against an expected set of values

The noise source tests and FIPS 140-2 KATs can also be run on demand by the user and occur automatically at various times during TOE operation.

Guidance	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.

The 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) states that if one or more of the self-tests fail, the FortiGate unit switches to FIPS Error mode. The unit shuts down all interfaces including the console and blocks traffic.

To resume normal FIPS-CC mode operation, the administrator must power cycle the unit. If the self-tests pass after the reboot, the unit will resume normal FIPS-CC operation. If a self-test continues to fail after rebooting, there is likely a serious firmware or hardware problem and the unit should be removed from the network until the problem is solved.

Testing	Future versions of this cPP will mandate a clearly defined minimum set of self-tests. But also for this version of the cPP it is expected that at least the following tests are performed: • Verification of the integrity of the firmware and executable software of the TOE; and • 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: • FIPS 140-2, chap. 4.9.1, Software/firmware integrity test for the verification of the integrity of the firmware and executable software. • 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 verify that the self-tests described above are either carried out during initial start-up and that the developer has justified any deviation from this (if applicable).

The tester observed the TOE output during start up and confirmed that it performed the self-tests specified in the Security Target.

2.6.4 FPT_TUD_EXT.1 Trusted Update

TSS	The evaluator shall verify that the TSS describes all TSF software update mechanisms for updating the system software. The evaluator shall verify that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails. Alternatively an approach using a published hash can be used. In this case the TSS shall detail this mechanism instead of the digital signature verification mechanism. The evaluator shall verify that the TSS describes the method by which the digital signature or published hash is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature or published hash of the update, and the actions that take place for both successful and unsuccessful signature verification or published hash verification. If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the TSS contains a description of how the certificates are contained on the device. The evaluator also ensures that the TSS (or guidance documentation) describes how the certificates are installed/updated/selected, if necessary.

The administrator may query the current version of the TOE via the GUI or CLI. The TOE will notify administrators if a new update file is available, but the update process will not commence until requested by the administrator.

Updates to the TOE are applied in accordance with the following process:

- The administrator downloads the upgrade image/package from the Fortinet website.
- Once downloaded, the administrator must transfer the image to the TOE via a trusted path (e.g. the web interface).

- Upon initiating the update process, the TOE will attempt to verify the integrity and authenticity of the update package. This is achieved via the verification of a 2048-bit RSA signature that is applied to the package by the Fortinet development team.
    - If the signature cannot be verified, or the integrity of the package cannot be confirmed, the upgrade will fail and an audit log generated accordingly.
    - If the signature is verified correctly and the integrity of the package is confirmed, the upgrade will be applied and the TOE restarted.

Additionally, MD5 hashes of each update file are published along with the image on the Fortinet website. Administrators may compare these published hashes against the hashes of the file they have downloaded to ensure that the file is valid.

Guidance	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.

The 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) contains instructions on how to download TOE updates and verify their integrity using an MD5 hash.

Testing	The evaluator shall perform the following tests: • The evaluator performs the version verification activity to determine the current version of the product as well as the most recently installed version (should be the same version before updating). • 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. 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 (if digital signatures are used) or a version that does not match the published hash (if published hashes are used) 2. An image that has not been signed (if digital signatures are used) or an image without published hash (if published hashes are used) 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) (only if digital signatures are used). 4. 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. The evaluator shall perform the above tests for all methods supported (manual updates, automatic checking for updates, automatic updates).

The evaluators obtained a genuine update file via the guidance procedures and applied the update to the TOE. The process executed as expected and the *show version* command indicated the new version of the TOE post-upgrade.

The evaluators modified the update file using a hex editor and attempted to install it on the TOE. The evaluators confirmed that the upgrade process did not proceed and the TOE generated an audit log entry indicating the reason for the process failure.

The evaluators created an update file with an invalid signature and attempted to install it on the TOE. The evaluators confirmed that the upgrade process did not proceed and the TOE generated an audit log entry indicating the reason for the process failure.

2.6.5 FPT_STM.1 Reliable Time Stamps

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

Per Section 8.12 of the ST , the TOE maintains its own time source which is free from outside interference. This time source is used for the purposes of generating audit logs and other time-sensitive operations on the TOE, including cryptographic key regeneration intervals. The TOE may also connect to an NTP server for synchronisation.

Guidance	The evaluator examines the guidance documentation to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the guidance documentation instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication.

Chapter 31 of the 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) contains instructions for configuring the time locally on the device as well as instructions on how to configure the TOE to connect to and sync time with an NTP server.

Testing	The evaluator shall perform the following tests: • Test 1: The evaluator uses the guidance documentation to set the time. The evaluator shall then use an available interface to observe that the time was set correctly. • Test 2: If the TOE supports the use of an NTP server; the evaluator shall use the guidance documentation to configure the NTP client on the TOE, and set up a communication path with the NTP server. The evaluator will observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the guidance documentation. If the audit component of the TOE consists of several parts with independent time information, then the evaluator shall verify that the time information between the different parts are either synchronized or that it is possible for all audit information to relate the time information of the different part to one base information unambiguously.

The evaluators manually set the date and time. The evaluators then observed the system date/time and confirmed that the value was as expected.

The evaluators misconfigured the TOE date/time and then configured the TOE to use NTP. After a period of time, evaluators examined the TOE date/time and confirmed that it had been updated by contacting the NTP server and the date/time was now correct.

2.6.6 FPT_FLS.1 Failure with preservation of secure state

TSS	The evaluator shall ensure the TSS describes how the TOE ensures a shutdown upon a self-test failure, a failed integrity check of the TSF executable image, or a failed health test of the noise source. If there are instances when a shut-down does not occur, e.g., a failure is deemed non-security relevant, those cases are identified and a rationale supporting the classification and justification why the TOE's ability to enforce its security policies is not affected.

The TOE performs the following self-tests upon initialisation:

- **CPU and Memory BIOS self-tests**

  CPU and memory are initialized by exercising a set of known answer tests and the BIOS is compared against a known checksum of the image. The memory is zeroized and then has a random pattern written and read from the memory.

- **Boot loader image verification**

  The boot loader will compare the image of the TOE to a known checksum of the image prior to booting.

- **Noise source tests**

  The noise source is started and pattern analysis is done on the output to ensure that the source is not stuck in a cryptographically weak state. These include both the repetition and adaptive proportion tests

- **FIPS 140-2 Known Answer Tests (KAT)**

  Comparison of a number of cryptographic functions against an expected set of values

The noise source tests and FIPS 140-2 KATs can also be run on demand by the user and occur automatically at various times during normal TOE operation.

The above tests ensure that the CPU and memory utilised by the TOE are functioning as intended, the BIOS and boot loader image are authentic and stable, the noise source used for entropy generation is functioning at capability and that the cryptographic algorithms used by the TOE are operating correctly. Together, these tests ensure that the TOE is operating at its intended level of capability.

There are several self-tests in which the TOE will enter an error-based blocking state. The first is a failure of any self-tests upon initialization of the TOE. This includes (but is not limited to) BIOS, software/firmware integrity checks and cryptographic self-tests. Upon the detection of one of these test failures, the TOE will halt and no further processing will occur until the TOE is reset.

Guidance	N/A

N/A

Testing	N/A

N/A

2.7 TOE Access (FTA)

2.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

TSS	N/A

N/A

Guidance	N/A

N/A

Testing	The evaluator shall perform the following test(s): • 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.

The evaluators configured the TOE in line with the guidance and established a local session. The evaluators confirmed that after the configured time period had expired, the session was closed by the TOE and re-authentication was required before access to TOE functions was again permitted.

The evaluators repeated the above test with a variety of time periods and confirmed that the TOE behaviour was identical for each defined threshold.

2.7.2 FTA_SSL.3 TSF-initiated Termination

TSS	N/A

N/A

Guidance	N/A

N/A

Testing	The evaluator shall perform the following test: • 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.

The evaluators configured the TOE in line with the guidance and established a local session. The evaluators confirmed that after the configured time period had expired, the session was closed by the TOE.

The evaluators repeated the above test with a variety of time periods and confirmed that the TOE behaviour was identical for each configured threshold.

2.7.3 FTA_SSL.4 User-initiated Termination

TSS	N/A

N/A

Guidance	N/A

N/A

Testing	The evaluator shall perform the following tests: a) The evaluator initiates an interactive local session with the TOE. The evaluator then follows the guidance documentation to exit or log off the session and observes that the session has been terminated. b) 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.

The evaluators initiated a local session with the TOE and manually ended the session using the commands listed in the operational guidance. The evaluators confirmed that the TOE terminated the session and re-authentication was required before TOE functions could be accessed.

The evaluators initiated a remote session with the TOE and confirmed that the behaviour was the same as for local sessions.

2.7.4 FTA_TAB.1 Default TOE Access Banners

TSS	The evaluator shall check the TSS to ensure that it details each method of access (local and remote) available to the administrator (e.g., serial port, SSH, HTTPS).

Per Section 8.13 of the ST, TOE administrators may access the TOE remotely (via the HTTPS/TLS web GUI or SSH) or locally (via the console port).

Users connecting to the TOE will be presented with a warning and consent banner prior to authentication.

Guidance	N/A

N/A

Testing	The evaluator shall also perform the following test: • 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.

Evaluators used the commands specified in the guidance documentation to set an access banner. Evaluators then connected to the TOE using both the remote and local channels and confirmed that the banner was displayed in both instances.

2.8 Trusted path/channels (FTP)

2.8.1 FTP_ITC.1 Inter-TSF trusted channel

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

Per Sections 8.2 and 8.14 of the ST, the TOE provides a trusted channel between itself and the following entities:

- Between the TOE and a FortiAnalyzer logging platform using TLS; and
- Between the TOE and VPN endpoints using IPsec.

These channels can be initiated by either the TOE or the authorised entities.

Administrators may utilise an IPsec tunnel on top of SSH or HTTPS when performing remote administration to provide additional transport security.

Guidance	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.

The IPsec VPN section of the FortiOS Handbook (Ref. [11]) contains instructions on establishing an IPsec VPN tunnel between the TOE and a peer. The troubleshooting sub-section of the IPsec VPN section of the handbook provides instructions on actions to take if an IPsec VPN tunnel fails.

The 'FIPS 140-2 and CC compliant operation for FortiOS 5.4' guide (Ref.[9]) contains instructions on establishing a connection to the external FortiAnalyzer device, how the TOE operates in the event that its connection to its FortiAnalyzer fails and the actions the administrator should take if the connection does not automatically recover.

Testing	The evaluator shall perform the following tests: a) The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful. b) For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be initiated from the TOE. c) The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext. d) The evaluators shall, for each protocol associated with each authorized IT entity tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected. Further assurance activities are associated with the specific protocols.

The evaluators established an IPsec tunnel between the TOE and an IPsec peer. The evaluators confirmed that the channel can be established between the TOE and an entity from the TOE.

The evaluators confirmed (via Wireshark) that the data sent between the TOE and the peer is encrypted and not sent in plaintext.

The evaluators interrupted the channel between the peer and the TOE (via manually unplugging the cable). Evaluators confirmed that manual intervention was required to re-establish the channel and no data was sent between the TOE and the IT entity until this action was taken.

The evaluators then performed the same tests on the connection between the TOE and the FortiAnalyzer external logging device. Evaluators confirmed that the connection could be established in accordance with the guidance documentation and that the connection was initiated by the TOE.

The connection between the TOE and the FortiAnalyzer was observed to be encrypted and no data was observed to be sent in plaintext. Upon physically interrupting and restoring the connection, the TOE re-established the connection. No data was sent as plaintext during this re-establishment phase.

2.8.2 FTP_TRP.1 Trusted Path

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

Per Section 8.4 of the ST, the TOE provides a trusted path between itself and remote administrative users using the following protocols:

- TLS (Versions 1.1 and 1.2) and HTTPS (in compliance with RFC 2818) for the Web GUI; and
- SSH in compliance with the following RFCs: 4251, 4252, 4253, 4254, an5647, 5656, 6187 and 6668.

These protocols implement cryptographic algorithms to provide data transport security and integrity, preventing unauthorised access to (or modification of) data sent between the TOE and remote administrative users.

The trusted channels can be initiated by either the TOE or the authorised entities. Administrators may utilise an IPsec tunnel on top of SSH or HTTPS when performing remote administration to provide additional transport security.

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

Chapter 4 (Authentication) of the FortiOS handbook (Ref. [11]) provides information on setting up authentication methods. Additionally, Chapter 2 (Getting Started) provides guidance on how to log onto and administer the TOE.

Testing	The evaluator shall perform the following tests: a) The evaluators shall ensure that communications using each specified (in the guidance documentation) remote administration method is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful. b) For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be initiated from the TOE. c) The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext. d) The evaluators shall ensure that, for each protocol associated with each authorized IT entity tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected. Further assurance activities are associated with the specific protocols.

The evaluators configured the TOE per the guidance documentation and successfully established a remote administration session via SSH, IPsec and HTTPS. The evaluators confirmed that there is no remote method of administration available that does not use the SSH, IPsec and/or HTTPS.

The evaluators confirmed (via Wireshark) that the traffic sent between the remote administrator and the TOE is encrypted using the appropriate/relevant protocol.

The evaluators confirmed (via modifying a packet) that the TOE detects modifications to traffic, rejects the packet and generates a syslog entry accordingly.

2.9 Firewall (FFW)

2.9.1 FFW_RUL_EXT.1 Stateful Traffic Filtering

TSS	The evaluator shall verify that the TSS provides a description of the TOE's initialization/startup process, which clearly indicates where processing of network packets begins to take place, and provides a discussion that supports the assertion that packets cannot flow during this process. The evaluator shall verify that the TSS also include a narrative that identifies the components (e.g., active entity such as a process or task) involved in processing the network packets and describe the safeguards that would prevent packets flowing through the TOE without applying the ruleset in the event of a component failure. This could include the failure of a component, such as a process being terminated, or a failure within a component, such as memory buffers full and cannot process packets.

The Fortinet family of appliances provide secure initialization procedure to ensure the integrity of the image and correct cryptographic functioning of the product prior to any information flowing.

The product starts from a powered down state and no signals on the wire. The device then powers on and undergoes the following initialization process:

- Bootstrap and Boot Loader
- Verification of the kernel, firmware and software images
- Loading and Initialization of
    - Kernel;
    - Firmware;

- Cryptographic known answer tests;
- Entropy gathering and DRBG initialization; and
- Cryptographic module

Once the kernel, firmware and cryptographic services have been initialized the TOE loads the configured firewall rules. The configuration file is then consulted and are initialized and configured with their network settings as specified and if appropriate transitioned to the link up sate. At this point packets may begin flowing through the various network interfaces. The CLI daemon is then started followed by the Web daemon – at this point, the TOE is ready to accept administrative connections.

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

N/A

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

The evaluators sent a stream of traffic at the TOE while it was initialising. The evaluators performed a Wireshark analysis and confirmed that no traffic was permitted to flow through the TOE while initialisation was in progress.

2.9.1.1 FFW_RUL_EXT.1.2/FFW_RUL_EXT.1.3/FFW_RUL_EXT.1.4

TSS	The evaluator shall verify that the TSS describes a stateful packet filtering policy and the following attributes are identified as being configurable within stateful traffic filtering rules for the associated protocols:
	• ICMPv4
	  ○ Type
	  ○ Code
	• ICMPv6
	  ○ Type
	  ○ Code
	• IPv4
	  ○ Source address
	  ○ Destination Address
	  ○ Transport Layer Protocol
	• IPv6
	  ○ Source address
	  ○ Destination Address
	  ○ Transport Layer Protocol and where defined by the ST author,
	  ▪ Extension Header Type, Extension Header Fields
	• TCP
	  ○ Source Port
	  ○ Destination Port
	• UDP
	  ○ Source Port
	  ○ Destination Port
	The evaluator shall verify that each rule can identify the following actions: permit or drop with the option to log the operation. The evaluator shall verify that the TSS identifies all interface types subject to the stateful packet filtering policy and explains how rules are associated with distinct network interfaces.

Section 8.16 of the ST section indicates that the TOE permits the configuration of stateful packet filtering policies.

The following protocols and associated attributes are configurable within each policy:

- ICMPv4 (RFC 792)
  - Type; and
  - Code
- ICMPv6 (RFC 4443)
  - Type; and
  - Code
- IPv4 (RFC 791)
  - Source address;
  - Destination Address; and
  - Transport Layer Protocol
- IPv6 (RFC 2460)
  - Source address;
  - Destination Address;
  - Transport Layer Protocol; and

- The following IPv6 Extension header types:
  - Hop-by-Hop Options;
  - Destination Options;
  - Routing;
  - Fragment;
  - Authentication Header; and
  - No Next Header.

- TCP (RFC 793)
  - Source Port; and
  - Destination Port
- UDP (RFC 768)
  - Source Port; and
  - Destination Port

Firewall rules can be configured to permit or drop traffic (with the generation of audit log entries for either option).

Each firewall rule can be tied to a specific interface (port1, wan1, etc.).

Guidance	The evaluators shall verify that the guidance documentation identifies the following attributes as being configurable within stateful traffic filtering rules for the associated protocols:
	• ICMPv4
	  ○ Type
	  ○ Code
	• ICMPv6
	  ○ Type
	  ○ Code
	• IPv4
	  ○ Source address
	  ○ Destination Address
	  ○ Transport Layer Protocol
	• IPv6
	  ○ Source address
	  ○ Destination Address
	  ○ Transport Layer Protocol and where defined by the ST author, Extension, Header Type, Extension Header Fields
	• TCP
	  ○ Source Port
	  ○ Destination Port
	• UDP
	  ○ Source Port
	  ○ Destination Port
	The evaluator shall verify that the guidance documentation indicates that each rule can identify the following actions: permit, drop, and log.
	The evaluator shall verify that the guidance documentation explains how rules are associated with distinct network interfaces.

The provided guidance documentation (Ref. [9], [10] and [11]) states that the above attributes are configurable as part of the firewall rules and each rule can configured to permit, drop and log.

Additionally the guidance material explains how to associate these rules with specific network interfaces.

Testing	The evaluator shall use the instructions in the guidance documentation to test that stateful packet filter firewall rules can be created that permit, drop, and log packets for each of the following attributes:
	• ICMPv4
	  ○ Type
	  ○ Code
	• ICMPv6
	  ○ Type
	  ○ Code
	• IPv4
	  ○ Source address
	  ○ Destination Address
	  ○ Transport Layer Protocol
	• IPv6
	  ○ Source address
	  ○ Destination Address
	  ○ Transport Layer Protocol and where defined by the ST author,
	  ▪ Extension Header Type, Extension Header Fields
	• TCP
	  ○ Source Port
	  ○ Destination Port
	• UDP
	  ○ Source Port
	  ○ Destination Port
	Repeat the test assurance activity above to ensure that stateful traffic filtering rules can be defined for each distinct network interface type supported by the TOE

The evaluators designed a number of test cases to exercise all of the above protocols and attributes for permit, deny and log actions. The evaluators performed these tests on all interface types provided by the TOE and confirmed that the TOE behaved as expected.

2.9.1.2 FFW_RUL_EXT.1.5

TSS	The evaluator shall verify that the TSS identifies the protocols that support stateful session handling. The TSS shall identify TCP, UDP, and ICMP if selected by the ST author. The evaluator shall verify that the TSS describes how stateful sessions are established (including handshake processing) and maintained. The evaluator shall verify that for TCP, the TSS identifies and describes the use of the following attributes in session determination: source and destination addresses, source and destination ports, sequence number, and individual flags. The evaluator shall verify that for UDP, the TSS identifies and describes the following attributes in session determination: source and destination addresses, source and destination ports. The evaluator shall verify that for ICMP (if selected), the TSS identifies and describes the following attributes in session determination: source and destination addresses, other attributes chosen in FFW_RUL_EXT.1.5. The evaluator shall verify that the TSS describes how established stateful sessions are removed. The TSS shall describe how connections are removed for each protocol based on normal completion and/or timeout conditions. The TSS shall also indicate when session removal becomes effective (e.g., before the next packet that might match the session is processed).

The TOE utilises a session database to track active sessions for TCP, UDP and ICMP (amongst other protocols). A number of variables (such as source/destination address and ports, sequence numbers, flags and TTL values) are utilised in the management of sessions. Incoming packets are inspected against the session database. Sessions that match all the security attributes and do not exceed the TTL are automatically passed on to their destination.

Periodically old sessions exceeding their TTL are removed from the database.

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

The provided guidance documentation (Ref. [9], [10] and [11]) describes how the TOE handles stateful session logging. Each session is granted a "session-ID" by the TOE which is used to log the data sent as part of an open session. When part of a session individual packets will not be logged however as part of the session close log total packets and data transmitted will be logged.

Testing	The evaluator shall verify that the guidance documentation describes stateful session behaviours. For example, a TOE might not log packets that are permitted as part of an existing session. The evaluator shall terminate the TCP session established per Test 1 as described in the TSS. The evaluator shall then immediately send a packet matching the former session definition in order to ensure it is not forwarded through the TOE without being subject to the ruleset. The evaluator shall terminate the TCP session established per Test 1 as described in the TSS. The evaluator shall then immediately send a packet matching the former session definition in order to ensure it is not forwarded through the TOE without being subject to the ruleset. The evaluator shall configure the TOE to permit and log UDP traffic. The evaluator shall establish a UDP session. Once a UDP session is established, the evaluator shall alter each of the session determining attributes (source and destination addresses, source and destination ports) one at a time in order to verify that the altered packets are not accepted as part of the established session. The evaluator shall expire (i.e., reach timeout) the UDP session established per Test 4 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset. If ICMP is selected, the evaluator shall configure the TOE to permit and log ICMP traffic. The evaluator shall establish a session for ICMP as defined in the TSS. Once an ICMP session is established, the evaluator shall alter each of the session determining attributes (source and destination addresses, other attributes chosen in FFW_RUL_EXT.1.5) one at a time in order to verify that the altered packets are not accepted as part of the established session. If applicable, the evaluator shall terminate the ICMP session established per Test 6 as described in the TSS. The evaluator shall then immediately send a packet matching the former session definition in order to ensure it is not forwarded through the TOE without being subject to the ruleset. The evaluator shall expire (i.e., reach timeout) the ICMP session established per Test 6 as described in the TSS. The evaluator shall then send a packet matching the former session in order to ensure it is not forwarded through the TOE without being subject to the ruleset.

The evaluators devised tests to permit and log TCP, UDP and ICMP.

While sessions for each protocol were established, the evaluator attempted to send traffic that did not match the existing session characteristics. The TOE did not permit this traffic as it did not match any existing session(s).

The evaluators expired/terminated each session and attempted to send traffic matching the former session definition. The evaluators confirmed that the TOE rejected these packets.

2.9.1.3 FFW_RUL_EXT.1.6

TSS	The evaluator shall verify that the TSS identifies the following as packets that will be automatically dropped and are counted or logged: a) Packets which are invalid fragments, including a description of what constitutes an invalid fragment b) Fragments that cannot be completely re-assembled c) Packets where the source address is defined as being on a broadcast network d) Packets where the source address is defined as being on a multicast network e) Packets where the source address is defined as being a loopback address f) The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address "reserved for future use" (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4; g) The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6; h) Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified i) Other packets defined in FFW_RUL_EXT.1.6

When encountered by the TOE, the following packets will be automatically dropped and an audit log generated for each event:

- Packets which are invalid fragments;
- Fragments that cannot be completely re-assembled;
- Packets where the source address is defined as being on a broadcast network;
- Packets where the source address is defined as being on a multicast network;
- Packets where the source address is defined as being a loopback address;
- Packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address "reserved for future use" (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
- Packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6; and
- Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified.

Guidance	The evaluator shall verify that the guidance documentation describes packets that are discarded and potentially logged by default. If applicable protocols are identified, their descriptions need to be consistent with the TSS. If logging is configurable, the evaluator shall verify that applicable instructions are provided to configure auditing of automatically rejected packets.

The 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) provides guidance on what configuration is needed to ensure that the TOE rejects and logs all the packets identified above.

Testing	The evaluator shall test each of the conditions for automatic packet rejection in turn. In each case, the TOE should be configured to allow all network traffic and the evaluator shall generate a packet or packet fragment that is to be rejected. The evaluator shall use packet captures to ensure that the unallowable packet or packet fragment is not passed through the TOE. For each of the cases above, the evaluator shall use any applicable guidance to enable dropped packet logging or counting. In each case above, the evaluator shall ensure that the rejected packet or packet fragment was recorded (either logged or an appropriate counter incremented).

The evaluators generated numerous tests to exercise each of the default reject rules. The evaluators confirmed that, even when all traffic is permitted, traffic matching the default reject rules was still dropped by the TOE.

The evaluators confirmed that the TOE logged each event appropriately.

2.9.1.4 FFW_RUL_EXT.1.7

TSS	The evaluator shall verify that the TSS explains how the following traffic can be dropped and counted or logged: a) Packets where the source address is equal to the address of the network interface where the network packet was received b) Packets where the source or destination address of the network packet is a link-local address c) Packets where the source address does not belong to the networks associated with the network interface where the network packet was received, including a description of how the TOE determines whether a source address belongs to a network associated with a given network interface

When encountered by the TOE, the following packets will be automatically dropped and an audit log generated for each event:

- Packets where the source address is equal to the address of the network interface where the network packet was received;
- Packets where the source or destination address of the network packet is a link-local address; and
- Packets where the source address does not belong to the networks associated with the network interface where the network packet was received, including a description of how the TOE determines whether a source address belongs to a network associated with a given network interface.

Guidance	The evaluator shall verify that the guidance documentation describes how the TOE can be configured to implement the required rules. If logging is configurable, the evaluator shall verify that applicable instructions are provided to configure auditing of automatically rejected packets.

The provided guidance documentation (Ref. [9], [10] and [11]) contains instructions on how to enable the dropping of the following types of traffic, local link traffic (address block 169.254.1.0 through 169.254.254.255), Class E traffic (240.0.0.0/4 and restrict the IPv6 address space to the allocated global unicast space.

All dropped packets are automatically logged when running in FIPS/CC mode.

Testing	The evaluator shall configure the TOE to drop and log network traffic where the source address of the packet matches that of the TOE network interface upon which the traffic was received. The evaluator shall generate suitable network traffic to match the configured rule and verify that the traffic is dropped and a log message generated.

The evaluator configured the TOE to drop and log network traffic where the source address of the packet matched that of the TOE network interface upon which the traffic was received. The evaluators generated traffic matching this rule and observed that it was dropped and logged by the TOE.

2.9.1.5 FFW_RUL_EXT.1.8

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

Packet rules are enforced in the order defined by the administrator. If no matching rule is found, the TOE will automatically deny the packets and generate a log entry accordingly.

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

The provided guidance documentation (Ref. [9], [10] and [11]) provides detailed instruction on how to configure the firewall and explains how individual packets are handled by FortiOS in detail as part of chapter 9 of the Handbook.

Testing	The evaluator shall devise two equal stateful traffic filtering rules with alternate operations – permit and drop. The rules should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first rule is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation. The evaluator shall repeat the procedure above, except that the two rules should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.

The evaluators configured a number of security policies to test the Permit/Drop functionality of the TOE. The evaluators confirmed that the TOE reacted as expected and that appropriate audit log entries were generated.

The evaluators repeated the previous test using a subset-based configuration. The evaluators confirmed that the TOE enforced the rules in the order defined by the administrator.

2.9.1.6 FFW_RUL_EXT.1.9

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

Packet rules are enforced in the order defined by the administrator. If no matching rule is found, the TOE will automatically deny the packets and generate a log entry accordingly.

Guidance	The evaluator shall verify that the guidance documentation describes the behavior if no rules or special conditions apply to the network traffic. If the behavior is configurable, the evaluator shall verify that the guidance documentation provides the appropriate instructions to configure the behavior to deny packets with no matching rules.

Packets without applicable rules are dropped by FortiOS by default and as such no configuration is necessary.

Testing	For each attribute in FFW_RUL_EXT.1.2, the evaluator shall construct a test to demonstrate that the TOE can correctly compare the attribute from the packet header to the ruleset, and shall demonstrate both the permit and deny for each case. The evaluator shall check the log in each case to confirm that the relevant rule was applied. The evaluator shall record a packet capture for each test to demonstrate the correct TOE behaviour.

The evaluators configured a number of traffic policies that met the requirements of this test. The evaluators sent traffic through the TOE that matched the Accept rule and confirmed that the TOE allowed the traffic to pass with no modification.

The evaluators then constructed a "junk" packet that did not match any of the policies in place. The evaluators confirmed that the default "deny all" policy was enforced and the TOE automatically dropped and logged the traffic flow.

2.9.1.7 FFW_RUL_EXT.1.10

TSS	The evaluator shall verify that the TSS describes how the TOE tracks and maintains information relating to the number of half-open TCP connections. The TSS should identify how the TOE behaves when the administratively defined limit is reached and should describe under what circumstances stale half-open connections are removed (e.g. after a timer expires).

The TOE maintains half-open TCP sessions in the same manner as full TCP sessions. Once the administrator-defined limit for total sessions is met, sessions (both valid and half-open) are automatically closed based on their timeout value (if not cleared manually by an administrator).

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

The provided guidance documentation (Ref. [9], [10] and [11]) provides detailed as part of the IPv4 DoS policy settings on how to configure the TOE to restrict TCP sessions.

Testing	The evaluator shall define a TCP half-open connection limit on the TOE. The evaluator shall generate TCP SYN requests to pass through the TOE to the target system using a randomised source IP address and common destination IP address. The number of SYN requests should exceed the TCP half-open threshold defined on the TOE. TCP SYN-ACK messages should not be acknowledged. The evaluator shall verify through packet capture that once the defined TCP half-open threshold has been reached, subsequent TCP SYN packets are not transmitted to the target system. The evaluator shall verify that when the configured threshold is reached that, depending upon the selection, either a log entry is generated or a counter is incremented.

The evaluators configured the IPv4 DoS policy settings on the TOE to restrict the number of half-open TCP connections permitted by the TOE. The evaluators then generated (using a script) a number of open connections that exceeded the limit and observed that they were denied and logged by the TOE.

3 VPNEP - SFR ASSURANCE ACTIVITIES

This section of the AAR defines each of the SFRs specified in the ST (Ref[8]), their corresponding assurance activities and the evaluator's findings in each case.

3.1 Audit Data Generation (FAU)

TSS	The evaluator shall verify that the TSS describes how the TSF can be configured to log network traffic associated with applicable rules. Note that this activity should have been addressed with a combination of the TSS assurance activities for FPF_RUL_EXT.1. The evaluator shall verify that the TSS describes how the TOE behaves when one of its interfaces is overwhelmed by network traffic. It is acceptable for the TOE to drop packets that it cannot process, but under no circumstances is the TOE allowed to pass packets that do not satisfy a rule that allows the permit operation or belong to an allowed established session. It may not always be possible for the TOE to audit dropped packets due to implementation limitations. These limitations and circumstances in which the event of dropped packets is not audited shall be described in the TSS.

Section 8.16 of the ST describes how applicable rules can be configured and how each of these rules then apply/correspond to the generation of audit logs.

Additionally, Section 8.12 indicates that the TOE may receive traffic above the capacity of the interface – in this instance, it will drop all packets that cannot be processed. These events, including dropped packets, are logged by the TOE.

Guidance	The evaluator shall verify that the TSS describes how the TSF can be configured to log network traffic associated with applicable rules. Note that this activity should have been addressed with a combination of the TSS assurance activities for FPF_RUL_EXT.1. The evaluator shall verify that the TSS describes how the TOE behaves when one of its interfaces is overwhelmed by network traffic. It is acceptable for the TOE to drop packets that it cannot process, but under no circumstances is the TOE allowed to pass packets that do not satisfy a rule that allows the permit operation or belong to an allowed established session. It may not always be possible for the TOE to audit dropped packets due to implementation limitations. These limitations and circumstances in which the event of dropped packets is not audited shall be described in the TSS.

The FortiOS Log Reference (Ref. [10]) provides detailed information on all logs generated by the TOE and covers each field mandated by the cPP as well as providing additional information on many additional fields.

Additionally, the 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) provides instructions that need to be followed in order for the TOE to meet the logging requirements of the cPP.

Testing	The following test is expected to execute outside the context of the other requirements. While testing the TOE's compliance against the SFRs, either specific tests are developed and run in the context of this SFR or, as is typically done, the audit capability is turned on while testing the TOE's behavior in complying with the other SFRs in this EP. The evaluator shall attempt to flood the TOE with network packets such that the TOE will be unable to process all the packets. This may require the evaluator to configure the TOE to limit the bandwidth the TOE is capable to handling (e.g., use of a 10 MB interface). The evaluator shall then review the audit logs to verify that the TOE correctly records that it is unable to process all of the received packets and verify that the TOE logging behavior is consistent with the TSS.

Throughout the testing performed, the evaluators examined the TOE to determine whether audit log entries for all auditable events were generated. The evaluators confirmed that all auditable events specified in the ST are generated.

The evaluators confirmed that, in the event that the TOE is subjected to more traffic than the interfaces are able to handle, that the TOE automatically drops all received packets until the overwhelming traffic ceases. The TOE generates audit logs for these events appropriately.

3.2 Security Management (FMT)

3.2.1 FMT_SMF.1 Specification of Management Functions

TSS	The evaluator shall verify that the TSS describes how the traffic filter rules for VPN traffic can be configured. Note that this activity can be addressed in parallel with the TSS assurance activities for FPF_RUL_EXT.1.

Per Section 8.11 of the ST, the TOE defines a single role, which is that of the Security Administrator. The Security Administrator is able to perform the following functions:

- Administer the TOE locally and remotely;
- Configure the access banner;
- Configure the session inactivity time before session termination or locking;
- Update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- Configure the cryptographic functionality;
- Modify, delete, generate and/or import cryptographic keys;
- Configure the IPsec functionality;
- Import X.509v3 certificates;
- Enable, disable, determine and modify the behavior of all the security functions of the TOE identified in this EP to the Administrator; and
- Ability to configure firewall rules.

Guidance	The evaluator shall verify that the operational guidance describes how to configure the traffic filter rules, including how to set any configurable defaults and how to configure each of the applicable rule attributes, actions, and associated interfaces. The evaluator must ensure that the operational guidance also provides instruction that would allow an administrator to ensure that configured rules are properly ordered. Note that this activity should have been addressed with the Guidance assurance activities for FPF_RUL_EXT.1.

The provided guidance documentation (Ref. [9], [10] and [11]) indicates that the attributes, actions and interfaces specified in this requirement are configurable as part of the firewall rules and each rule can be configured to permit, drop and log.

The guidance material explains how to associate these rules with specific network interfaces.

Testing	The evaluator shall devise tests that demonstrate that the functions used to configure the TSF yield expected changes in the rules and that they are correctly enforced. A number of rule combination and ordering scenarios need to be configured and tested by attempting to pass both valid and invalid network traffic through the TOE. Note that this activity should have been addressed with a combination of the Test assurance activities for FPF_RUL_EXT.1

This testing was carried out as part of FPF_RUL_EXT.1.

3.3 Cryptographic Support (FCS)

3.3.1 FCS_CKM.1/IKE Cryptographic Key Generation (for IKE Peer Authentication)

TSS	The evaluator shall check to ensure that the TSS describes how the key-pairs are generated. In order to show that the TSF implementation complies with FIPS PUB 186-4, the evaluator shall ensure that the TSS contains the following information: • The TSS shall list all sections of Appendix B to which the TOE complies. • For each applicable section listed in the TSS, for all statements that are not "shall" (that is, "shall not", "should", and "should not"), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "shall not" or "should not" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE. • For each applicable section of Appendix B, any omission of functionality related to "shall" or "should" statements shall be described. Any TOE-specific extensions, processing that is not included in the Appendices, or alternative implementations allowed by the Appendices that may impact the security requirements the TOE is to enforce shall be described.

Table 14 (within Section 8.3 of the ST) indicates that the TOE generates asymmetric cryptographic keys used for IKE peer authentication in accordance with a specified cryptographic key generation algorithm in 186-4.

This table lists all sections of Appendix B to which the TOE complies. There are no "shall [not]" or "should [not]" functionality requirements claimed in this section.

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

The "IPsec VPN in the web-based manager" section of the FortiOS Handbook (Ref. [11]) provides instructions on configuration of IKE key generation for the use in IPsec VPN tunnels.

Testing	The evaluator shall use the key pair generation portions of "The FIPS 186-4 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)" and "The RSA Validation System (RSA2VS)" as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

The key cryptographic functionality implemented within the TOE has been given CAVP certificates - these certificate numbers are listed with the ST (Ref. [8]).

3.4 Identification and Authorization (FIA)

3.4.1 FIA_AFL.1 Authentication Failure Handling

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

Per Section 8.8 of the ST, the TOE permits administrators to set a positive integer for failed remote authentication attempts. When this limit is met, the remote user must wait for a defined period of time before further authentication attempts can be made.

Administrators connecting via a local connection (console) or remote (HTTPS/TLS or SSH) must provide a valid username and password to complete authentication. If the username and password provided is incorrect, the administrator is presented with an error.

Guidance	The evaluator shall also examine the operational guidance to ensure that instructions for configuring the number of successive unsuccessful authentication attempts (FIA_AFL.1.1) and time period (FIA_AFL.1.2, if implemented) are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each "action" specified (if that option is chosen). If different actions or mechanisms are implemented depending on the secure protocol employed (e.g., TLS vs. SSH), all must be described.

The evaluators configured the TOE in line with the provided guidance and set a defined number of unsuccessful authentication attempts to be permitted. The evaluators confirmed that once this limit is reached, further authentication attempts are not successful.

The evaluators configured the time period before additional authentication attempts would be permitted and, once the time period lockout caused by the previous attempt had expired, confirmed that further authentication attempts were possible.

The evaluators performed the above tests for both SSH and TLS and confirmed that the behaviour was identical in both instances.

Testing	The evaluator shall perform the following tests for each method by which remote administrators access the TOE (e.g., TLS, SSH): • The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE. The evaluator shall test that once the limit is reached, attempts with valid credentials are not successful. For each action specified by the requirement, the evaluator shall show that following the operational guidance and performing each action to allow the remote administrator access are successful. • The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE and a time period after which valid logins will be allowed for a remote administrator. After exceeding the specified number of invalid login attempts and showing that valid login is not possible, the evaluator shall show that waiting for the interval defined by the time period before another access attempt will result in the ability for the remote administrator to successfully log on using valid credentials.

The evaluators configured the TOE in line with the provided guidance and set a defined number of unsuccessful authentication attempts to be permitted. The evaluators confirmed that once this limit is reached, further authentication attempts are not successful.

The evaluators configured the time period before additional authentication attempts would be permitted and once the time period lockout caused by the previous attempt had expired, confirmed that further authentication attempts were possible.

3.4.2 FIA_X509_EXT.4 X.509 Certificate Identity

TSS	The TSS shall describe all certificate stores implemented that contain certificates used to meet the requirements of this EP. This description shall contain information pertaining to how certificates are loaded into the store, and how the store is protected from unauthorized access. The TSS description will also include a discussion as to how the TOE forms a certification path as specified in the standard and how certificates are validated (CRL and/or OCSP are included in the discussion, as well as the certificate path validation algorithm).

Per Section 8.7, when using certificates for peer authentication, the TOE will only establish a trusted channel to peers that provide a valid certificate. The TOE will compare the DN of the peer against the DN stored in the associated certificate. If the two values are not a match, the TOE will not establish the connection.

The TOE provides a single certificate store contained within the TOE file system and no access is provided except via the web GUI or command line for addition/deletion of certificates or CRLs. Certificates must be enabled via the Features list before access to the store is provided.

Once a certificate is loaded into the store, the only editing allowed is to add a comment that is associated with the certificate. No other editing of certificates, keys or CRLs is allowed.

Certificate validation takes place during the handshake of HTTPS or IPsec VPN connections. The TOE validates certificates via Certificate Revocation List (CRL). In determining certificate validity, the following process is used:

- The remote client sends its certificate and associated key to the TOE.
- The TOE compares the received certificate and key against the certificate store (CA certificates, remote certificates, etc.) to determine that the certificate is authentic.

- The TOE then compares the certificate against any loaded CRLs for validity confirmation.
    - If the certificate is determined to be invalid or revoked, the certificate is rejected and the connection is not established
    - If the certificate is determined to be valid, the connection process continues.

Guidance	The evaluator shall verify that the operational guidance describes how to configure the TOE to either allow or disallow the establishment of an SA.

The "IPsec VPN in the web-based manager" section of the FortiOS Handbook (Ref. [11]) provides instructions on the configuration of SAs within IPsec connections.

Testing	This SFR is tested as part of FCS_IPSEC_EXT.1 as defined by the NDcPP.

N/A

3.5 Packet Filtering (FPF)

3.5.1 FPF_RUL_EXT.1 Rules for Packet Filtering

3.5.1.1 FPF_RUL_EXT.1.1

TSS	The evaluator shall verify that the TSS provide a description of the TOE's initialization/startup process, which clearly indicates where processing of network packets begins to take place, and provides a discussion that supports the assertion that packets cannot flow during this process. The evaluator shall verify that the TSS also includes a narrative that identifies the components (e.g., active entity such as a process or task) involved in processing the network packets and describes the safeguards that would prevent packets flowing through the TOE without applying the ruleset in the event of a component failure. This could include the failure of a component, such as a process being terminated, or a failure within a component, such as memory buffers full and cannot process packets.

The Fortinet family of appliances provides a secure initialization procedure to ensure the integrity of the image and correct cryptographic functioning of the product prior to any information flowing. The product starts from a powered down state and no signals on the wire. The device then powers on and undergoes the following initialization process:

- Bootstrap and Boot Loader
- Verification of the kernel, firmware and software images
- Loading and Initialization of
    - Kernel;
    - Firmware;
    - Cryptographic known answer tests;
    - Entropy gathering and DRBG initialization; and
    - Cryptographic module

Once the kernel, firmware and cryptographic services have been initialized the TOE loads the configured firewall rules. The configuration file is then consulted and initialized and configured with their network settings as specified and if appropriate transitioned to the link up state. At this point packets may begin flowing through the various network interfaces.

The CLI daemon is then started followed by the Web daemon. At this point, the TOE is ready to accept administrative connections.

Guidance	The operational guidance associated with this requirement is assessed in the subsequent test assurance activities.

N/A

Testing	The evaluator shall attempt to get network traffic to flow through the TOE while the TOE is being initialized. A steady flow of network packets that would otherwise be denied by the ruleset should be directed at the TOE's interfaces, with packet sniffers listening to see if any network traffic is allowed through.

The evaluators configured a peer device to send a constant flow of packets via the TOE to a machine on a separate subnet. The TOE was powered on and the evaluators confirmed (via Wireshark) that no packets flowed through the TOE to the target subnet while the initialisation process was underway.

3.5.1.2 FPF_RUL_EXT.1.2

TSS	The evaluator shall verify that the TSS indicates that the following protocols are supported: • RFC 791 (IPv4) • RFC 2460 (IPv6) • RFC 793 (TCP) • RFC 768 (UDP) The evaluator shall verify that the TSS describes how conformance with the identified RFCs has been determined by the TOE developer (e.g., third party interoperability testing, protocol compliance testing).

Per the Security Target, the TOE permits the configuration of stateful packet filtering policies. The following protocols are configurable within each policy:

- IPv4 (RFC 791);
- IPv6 (RFC 2460);
- TCP (RFC 793); and
- UDP (RFC 768).

Guidance	The evaluator shall verify that the operational guidance indicates that the following protocols are supported: • RFC 791 (IPv4) • RFC 2460 (IPv6) • RFC 793 (TCP) • RFC 768 (UDP) The guidance will describe the other protocols contained within the ST (e.g., IPsec, IKE, potentially HTTPS, SSH, and TLS) that are processed by the TOE. The evaluator ensures it is made clear what protocols were not considered as part of the TOE evaluation.

The provided guidance documentation (Ref. [9], [10] and [11]) states that the above protocols are configurable as part of firewall rules. The guidance material explains how to associate these rules with specific network interfaces.

Testing	The testing associated with this requirement is addressed in the subsequent test assurance activities.

N/A

3.5.1.3 FPF_RUL_EXT.1.5

TSS	The evaluator shall verify that the TSS describes a Packet Filtering policy and the following attributes are:
	• IPv4
	  ○ Source address
	  ○ Destination Address
	  ○ Protocol
	• IPv6
	  ○ Source address
	  ○ Destination Address
	  ○ Next Header (Protocol)
	• TCP
	  ○ Source Port
	  ○ Destination Port
	• UDP
	  ○ Source Port
	  ○ Destination Port
	The evaluator shall verify that each rule can identify the following actions: permit, deny, and log.
	The evaluator shall verify that the TSS identifies all interface types subject to the Packet Filtering policy and explains how rules are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used) they can be treated collectively as a distinct network interface.

Per Section 8.16 of the ST, each rule can be tied to a specific interface (port1, wan1, etc.). Each packet that arrives on an interface is subject to the enforcement of the stateful traffic filtering.

The following protocols and associated attributes are configurable within each policy:

- ICMPv4 (RFC 792)
  - Type; and
  - Code
- ICMPv6 (RFC 4443)
  - Type; and
  - Code
- IPv4 (RFC 791)
  - Source address;
  - Destination Address; and
  - Transport Layer Protocol
- IPv6 (RFC 2460)
  - Source address;
  - Destination Address;
  - Transport Layer Protocol; and
  - The following IPv6 Extension header types:
    - Hop-by-Hop Options;
    - Destination Options;
    - Routing;
    - Fragment;

        - Authentication Header; and
        - No Next Header.
- TCP (RFC 793)
    - Source Port; and
    - Destination Port
- UDP (RFC 768)
    - Source Port; and
    - Destination Port

Firewall rules can be configured to permit or drop traffic (with the generation of audit log entries for either option).

Guidance	The evaluators shall verify that the operational guidance identifies the following attributes as being configurable within Packet filtering rules for the associated protocols:
	• IPv4
	  ○ Source address
	  ○ Destination Address
	  ○ Protocol
	• IPv6
	  ○ Source address
	  ○ Destination Address
	  ○ Next Header (Protocol)
	• TCP
	  ○ Source Port
	  ○ Destination Port
	• UDP
	  ○ Source Port
	  ○ Destination Port
	The evaluator shall verify that the operational guidance indicates that each rule can identify the following actions: permit, deny, and log.
	The evaluator shall verify that the operational guidance explains how rules are associated with distinct network interfaces.

The provided guidance documentation (Ref. [9], [10] and [11]) indicates that the above attributes are configurable as part of the firewall rules and each rule can configured to permit, drop and log. The guidance material explains how to associate these rules with specific network interfaces.

Testing	The evaluator shall use the instructions in the operational guidance to test that packet filter rules can be created that permit, deny, and log packets for each of the following attributes:
	• IPv4
	  ○ Source address
	  ○ Destination Address
	  ○ Protocol
	• IPv6
	  ○ Source address
	  ○ Destination Address
	  ○ Next Header (Protocol)
	• TCP
	  ○ Source Port
	  ○ Destination Port
	• UDP
	  ○ Source Port
	  ○ Destination Port
	Repeat the test assurance activity above to ensure that Packet filtering rules can be defined for each distinct network interface type supported by the TOE.

The evaluators produced numerous TOE configurations to ensure that the TOE permits the use of rules for each protocol, attribute and action. The evaluators sent traffic through the TOE and confirmed that, for each combination of protocol, attribute and action, the TOE behaved as expected.

The evaluators repeated the tests to ensure that the rules were permitted for each interface type provided by the TOE.

3.5.1.4 FPF_RUL_EXT.1.6

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

Each packet that arrives on an interface is subject to the enforcement of the stateful traffic filtering. This filtering verifies if the connection is part of an established session or if it is a new connection. If the security attributes of the incoming connection request match those already present for an entry in the state table of the TOE the information flow is automatically allowed. Otherwise this is considered a new connection attempt.

For a new connection attempt a list of administrator-defined security rules are consulted in their sequence order until a match is found for that packet. The packet is then allowed, denied or dropped based on the configuration of this rule. The session database is consulted to see if an additional session can be created by examining how many currently exist in the database. If this number is below the hardware limit sessions are established by writing the attributes and a TTL into the session database.

If the connection is allowed, a new session is written into the list of established sessions and can be used to allow subsequent packets for this connection. If logging is enabled for the rule the audit event is sent in real time to the audit server. Any new session will have the first packet of the exchange inspected according to the firewall table as described above, such as the TCP SYN packet during a typical TCP session negotiation for both the sender and receiver. The TOE will write to the session table the expected source and destination ports for this communication flow based on the observed IP headers.

For FTP the initial handshake communication on port 21 for FTP will be inspected, as well as the server response indicating the expected data and control communication ports. A session will be written to the state table reflecting the expected source and destination ports based on this packet inspection.

For H.323 the TOE will inspect the ARQ request to the gatekeeper device and allow the establishment of this communication via an entry into the state table. The TOE will inspect the response from the gatekeeper to determine the expected UDP port and IP address of the device registered with the gatekeeper and write a session to the session table indicating that this communication is expected and should be allowed.

The TOE utilises a session database to track active sessions for TCP, UDP and ICMP (amongst other protocols). A number of variables (such as source/destination address and ports, sequence numbers, flags and TTL values) are utilised in the management of sessions. Periodically old sessions exceeding their TTL are removed from the database. Each FortiGate™ appliance has a pre-defined number of sessions it can track and is specified on the specifications sheet.

When encountered by the TOE, the following packets will be automatically dropped and an audit log generated for each event:

- Packets which are invalid fragments (see below);
- Fragments that cannot be completely re-assembled;
- Packets where the source address is defined as being on a broadcast network;
- Packets where the source address is defined as being on a multicast network;
- Packets where the source address is defined as being a loopback address;
- Packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address "reserved for future use" (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
- Packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;
- Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified.
- Packets where the source address is equal to the address of the network interface where the network packet was received;
- Packets where the source or destination address of the network packet is a linklocal address; and
- Packets where the source address does not belong to the networks associated with the network interface where the network packet was received, including a description of how the TOE determines whether a source address belongs to a network associated with a given network interface.

The TOE is capable of detecting fragmented packets. When fragmented packets arrive at their destination, they are reassembled and read. If the fragments do not arrive together, they must be held until all of the fragments arrive. Reassembly of a packet requires all of the fragments. The TOE in the evaluated configuration will attempt to reassemble fragmented packets. When these packets arrive at the TOE they will be held by the TOE for reassembly until the TTL expires.

Should the TOE detect that there is a missing or invalid fragment during the reassembly the packet will be dropped and logged. This behaviour is capable of being modified or overwritten by the TOE administrator.

Packet rules are enforced in the order defined by the administrator. If no matching rule is found, the TOE will automatically deny the packets and generate a log entry accordingly

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

Chapter 9 of the FortiOS handbook (Ref. [11]) provides instructions on how to configure and organise firewall rules and policies.

Testing	The evaluator shall perform the following tests: • The evaluator shall devise two equal Packet filtering rules with alternate operations – permit and deny. The rules should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first rule is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation. • The evaluator shall repeat the procedure above, except that the two rules should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.

The evaluators created a set of rules (permit and deny) and deployed the rules in alternating orders. The evaluators transmitted packets through the TOE and, via Wireshark and syslog, confirmed that the rules were enforced by the TOE in the order specified

The evaluators created two rules (one as a subset of another) and, as before, transmitted traffic and monitored the results. The evaluators confirmed that the first-in-line rule was always enforced.

3.5.1.5 FPF_RUL_EXT.1.7

TSS	The evaluator shall verify that the TSS describes the process for applying Packet filtering rules and also that the behavior (either by default, or as configured by the administrator) is to deny packets when there is no rule match unless another required conditions allows the network traffic (i.e., FPF_RUL_EXT.1.6 or FPF_RUL_EXT.1.7).

Each packet that arrives on an interface is subject to the enforcement of the stateful traffic filtering. Packet rules are enforced in the order defined by the administrator. If no matching rule is found, the TOE will automatically deny the packets and generate a log entry accordingly

Guidance	The evaluator shall verify that the operational guidance describes the behavior if no rules or special conditions apply to the network traffic. If the behavior is configurable, the evaluator shall verify that the operational guidance provides the appropriate instructions to configure the behavior to deny packets with no matching rules.

Packets without applicable rules are dropped by the TOE by default as such no configuration is required. Some additional configuration to ensure that all packets identified above are dropped is provided in the FIPS and CC configuration guide (Ref. [9]).

Testing	See the definition of FPF_RUL_EXT.1 in the VPNEP for a complete list of the tests relevant to this requirement.

The evaluators produced a variety of configurations that specified rules to permit and log, deny and log and to implement no rules for the ICMPv4, ICMPv6, IPv4, IPv6, TCP and UDP protocols and for all specific attributes as specified in the FWEP.

The evaluators monitored Wireshark and the TOE audit log for each test performed and confirmed that, in each case, the TOE functioned as expected – permitting and logging, denying and logging and automatically rejecting (due to an absence of matching ruleset) in each case as appropriate.

3.6 Protection of the TSF (FPT)

3.6.1 FPT_FLS.1/SelfTest Fail Secure (Self-test Failures)

TSS	The evaluator shall ensure the TSS describes how the TOE ensures a shutdown upon a self-test failure, a failed integrity check of the TSF executable image, or a failed health test of the noise source. If there are instances when a shut-down does not occur, e.g., a failure is deemed non-security relevant, those cases are identified and a rationale supporting the classification and justification why the TOE's ability to enforce its security policies is not affected.

There are several self-tests in which the TOE will enter an error-based blocking state if failure occurs - this includes (but is not limited to) BIOS, software/firmware integrity checks and cryptographic self-tests. Upon the detection of one of these test failures, the TOE will halt and no further processing will occur until the TOE is reset.

Guidance	N/A

N/A

Testing	N/A

N/A

3.7 Selection-Based Requirements

3.7.1 Pre-Shared Key Composition (FIA_PSK_EXT)

3.7.1.1 FIA_PSK_EXT.1

TSS	The evaluator shall examine the TSS to ensure that it identifies all protocols that allow both text-based and bit-based pre-shared keys, and states that text-based pre-shared keys of 22 characters are supported. For each protocol identified by the requirement, the evaluator shall confirm that the TSS states the conditioning that takes place to transform the text-based pre-shared key from the key sequence entered by the user (e.g., ASCII representation) to the bit string used by the protocol, and that this conditioning is consistent with the last selection in the FIA_PSK_EXT.1.3 requirement.

Per Section 8.7 of the ST, the TOE accepts both text-based and bit-based pre-shared keys for IPsec VPN connections. The TOE accepts keys of a length between 6 and 128 characters (including 22-character PSKs).

The TOE converts text-based pre-shared keys into an authentication value as per RFC 2409 for IKEv1 or RFC 4306 for IKEv2, using SHA-1 or the PRF that is configured as the hash algorithm for the IKE exchanges.

Guidance	The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on the composition of strong text-based pre-shared keys, and (if the selection indicates keys of various lengths can be entered) that it provides information on the merits of shorter or longer pre-shared keys. The guidance must specify the allowable characters for pre-shared keys, and that list must be a super-set of the list contained in FIA_PSK_EXT.1.2. The evaluator shall confirm the operational guidance contains instructions for either entering bit-based pre-shared keys for each protocol identified in the requirement, or generating a bit-based pre-shared key (or both). The evaluator shall also examine the TSS to ensure it describes the process by which the bit-based pre-shared keys are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG_EXT.1 in the base PP.

The 'FIPS 140-2 and CC compliant operation for FortiOS 5.4' guide (Ref. [9]) describes how enabling FIPS-CC mode changes the password requirements of the administrator account. Additionally, the password policy section of the FortiOS handbook (Ref. [11]) provides information on the configuration of password settings and gives recommendations on how passwords/PSKs should be composed.

Testing	The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE). Note that one or more of these tests can be performed with a single test case. • The evaluator shall compose a pre-shared key of 22 characters that contains a combination of the allowed characters in accordance with the operational guidance, and demonstrates that a successful protocol negotiation can be performed with the key. • **Conditional**: If the TOE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length; the maximum length; and an invalid length. The minimum and maximum length tests should be successful, and the invalid length must be rejected by the TOE. • **Conditional**: If the TOE does not generate bit-based pre-shared keys, the evaluator shall obtain a bit-based pre-shared key of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key. • **Conditional**: If the TOE does generate bit-based pre-shared keys, the evaluator shall generate a bit-based pre-shared key of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.

The evaluators composed a list of 15+ PSKs of 22 character lengths that used combinations of all allowed characters defined in FIA_PSK_EXT.1. The evaluators confirmed that protocol negotiation with the TOE was successful with these keys.

The evaluators used keys of length 1, 22, 255 and 256 characters and confirmed that all keys except the 256-character key were accepted.

The TOE obtained a bit-based key of sufficient length and confirmed that protocol negotiation was possible using this key.

4 IPSEP - SFR ASSURANCE ACTIVITIES

This section of the AAR defines each of the SFRs specified in the ST (Ref. [8]), their corresponding assurance activities and the evaluator's findings in each case.

4.1 Audit Data Generation (FAU)

4.1.1 FAU_GEN.1.2/IPS Refinement Audit Data Generation (IPS)

TSS	The evaluator shall verify that the TSS describes how the TOE can be configured to log IPS data associated with applicable policies. The evaluator shall verify that the TSS describes what (similar) IPS event types the TOE will combine into a single audit record along with the conditions (e.g., thresholds and time periods) for so doing. The TSS shall also describe to what extent (if any) that may be configurable. For IPS_SBD_EXT.1, for each field, the evaluator shall verify that the TSS describes how the field is inspected and if logging is not applicable, any other mechanism such as counting that is deployed.

The TOE generates audit records for the following events:

- Seeding from entropy token or CP9, failure to seed or reseeding events;
- Start-up and shut-down of the IPS functions;
- All IPS auditable events;
- All dissimilar IPS events;
- All dissimilar IPS reactions;
- Totals of similar events occurring within a specified time period;
- Totals of similar reactions occurring within a specified time period; and
- All specifically defined auditable events listed in Table 11 – SFRs and associated auditable events and Table 12 – SFRs and associated auditable events (IPS).

For each auditable event, the TOE records the date and time of the event, subject identity (i.e. administrative user), type of event and/or reaction and (where applicable) the success or failure of the event.

Guidance	The evaluator shall verify that the operational guidance describes how to configure the TOE to result in applicable IPS data logging. The evaluator shall verify that the operational guidance provides instructions for any configuration that may be done in regard to logging similar events (e.g., setting thresholds, defining time windows, etc.).

The FortiOS Log Reference (Ref. [10]) provides detailed information on all logs generated by the TOE and covers each field mandated by the cPP as well as providing additional information on many additional fields. The 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) provides instructions that need to be followed in order for the TOE to meet the logging requirements of the cPP.

Testing	The evaluator shall test that the interfaces used to configure the IPS polices yield expected IPS data in association with the IPS policies. A number of IPS policy combination and ordering scenarios need to be configured and tested by attempting to pass both allowed and anomalous network traffic matching configured IPS policies in order to trigger all required IPS events. Note that this activity should have been addressed with a combination of the Test assurance activities for the other IPS requirements.

The evaluators performed a variety of tests that exercised all relevant IPS events. The evaluators confirmed that the behaviour of the TOE and the audit data generated was consistent with expectations.

4.2 Security Management (FMT)

4.2.1 FMT_SMF.1.1/IPS Specification of Management Functions (IPS)

TSS	The evaluator shall verify that the TSS describes how the IPS data analysis and reactions can be configured. Note that this activity should have been addressed with the TSS assurance activities for IPS_ABD_EXT.1, IPS_IPB_EXT.1 and IPS_ABD_EXT.1

The Security Administrator is able to perform the following functions relevant to the configuration of IPS analysis and reactions:

- Enable, disable signatures applied to sensor interfaces, and determine the behavior of IPS functionality;
- Modify these parameters that define the network traffic to be collected and analyzed:
  - Source IP addresses (host address and network address)
  - Destination IP addresses (host address and network address)
  - Source port (TCP and UDP)
  - Destination port (TCP and UDP)
  - Protocol (IPv4 and IPv6)
  - ICMP type and code
- Update (import) signatures;
- Create custom signatures;
- Configure anomaly detection;
- Enable and disable actions to be taken when signature or anomaly matches are detected;
- Modify thresholds that trigger IPS reactions;
- Modify the duration of traffic blocking actions;
- Modify the known-good and known-bad lists (of IP addresses or address ranges); and
- Configure the known-good and known-bad lists to override signature-based IPS policies.

Guidance	The evaluator shall verify that the operational guidance describes the instructions for each function defined in the SFR, describes how to configure the IPS data analysis and reactions, including how to set any configurable defaults and how to configure each of the applicable analysis pattern matching methods and reaction modes.

The Firewall section of the FortiOS handbook (Ref. [11]) contains the instructions on configuring IPS data analysis and reactions. The FIPS and CC Configuration guide (Ref. [9]) also contains some additional default configuration required by the TOE.

Testing	The evaluator shall perform the following tests: • The evaluator shall use the operational guidance to create a signature and enable it on an interface. The evaluator shall then generate traffic that would be successfully triggered by the signature. The evaluator should observe the TOE applying the corresponding reaction in the signature. • The evaluator shall then disable the signature and attempt to regenerate the same traffic and ensure that the TOE allows the traffic to pass with no reaction. • The evaluator shall use the operational guidance to import signatures and repeat the test conducted in Test 1. Note that all other functions should have been address with a combination of the test assurance activities for IPS_ABD_EXT.1 and IPS_SBD_EXT.1.

The evaluators configured, enabled and tested an IPS signature on an interface. The evaluator generated traffic matching the signature and confirmed that the TOE denied the traffic flow and generated appropriate log entries.

The evaluator then disabled the IPS signature and confirmed that the TOE allowed the traffic to flow with no reaction.

The evaluators imported a signature and repeated the test – the behaviour of the TOE was consistent in both cases.

4.3 Intrusion Prevention (IPS)

4.3.1 IPS_ABD_EXT.1.3 Anomaly-Based IPS Functionality

TSS	The evaluator shall verify that the TSS describes the composition, construction, and application of baselines or anomaly-based attributes specified in IPS_ABD_EXT.1.1. The evaluator shall verify that the TSS provides a description of how baselines are defined and implemented by the TOE, or a description of how anomaly-based rules are defined and configured by the administrator. The evaluator shall verify that each baseline or anomaly-based rule can be associated with a reaction specified in IPS_ABD_EXT.1.3. The evaluator shall verify that the TSS identifies all interface types capable of applying baseline or anomaly-based rules and explains how they are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used) they can be treated collectively as a distinct network interface.

The TOE supports the configuration of baseline and anomaly-based attributes when constructing signatures for inclusion in IPS policies. The following parameters may be included as part of the definition of baselines/anomalous traffic patterns in the format indicated above:

- Throughput (number of bytes or packets per time period (seconds/minutes/hours);
- Time of day;
- Frequency; and
- Thresholds.

In addition to the above, the TOE permits the use of the following header fields and data payloads for each supported protocol:

- IPv4: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; and IP Options.
- IPv6: Version; traffic class; flow label; payload length; next header; hop limit; source address; destination address; routing header; home address options.
- ICMP: Type; Code; Header Checksum; and Rest of Header (varies based on the ICMP type and code.
    - ICMPv4 data: characters beyond the first 4 bytes of the ICMP header.
- ICMPv6: Type; Code; and Header Checksum.
    - ICMPv6 data: characters beyond the first 4 bytes of the ICMP header.
- TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options.
    - TCP data (characters beyond the 20 byte TCP header), with support for detection of:
- FTP (file transfer) commands: help, noop, stat, syst, user, abort, acct, allo, appe, cdup, cwd, dele, list, mkd, mode, nlst, pass, pasv, port, pass, quit, rein, rest, retr, rmd, rnfr, rnto, site, smnt, stor, stou, stru, and type.
- HTTP (web) commands and content: commands including GET and POST, and administrator-defined strings to match URLs/URIs, and web page content; and
- SMTP (email) states: start state, SMTP commands state, mail header state, mail body state, abort state.
- UDP: Source port; destination port; length; and UDP checksum.
    - UDP data: characters beyond the first 8 bytes of the UDP header.

The TOE supports stream reassembly and is capable of detecting malicious payloads even if they are split across multiple non-fragmented packets. Each IPS rule can be configured with a reaction the TOE should take when a match is identified once added to an IPS Policy – allowing the traffic flow, denying the traffic flow or sending a TCP reset to the traffic source.

On some hardware models ports are tagged with names such as "WiFi", "LAN", "WAN" or "DMZ" either through identification on the user interface or through screened graphics on the TOE hardware. Tagging of the interfaces in this manner does not affect the TOE's capability for the enforcement of SFRs as all rules are capable of being configured on all types of interface. Identifying interfaces in this manner is done for the purposes of simplifying administration and identifying the correct port as well as applying a suitable default configuration.

Details as to the default configuration for each interface type as well as the methods for modifying their configuration can be found in the administrative guidance.

Guidance	The evaluator shall verify that the operational guidance provides instructions to manually create baselines or anomaly-based rules according to the selections made in IPS_ABD_EXT.1.1. Note that dynamic "profiling" of a network to establish a baseline is outside the scope of this PP. The evaluator shall verify that the operational guidance provides instructions to associate reactions specified in IPS_ABD_EXT.1.3 with baselines or anomaly-based rules. The evaluator shall verify that the operational guidance provides instructions to associate the different policies with distinct network interfaces.

The provided guidance documentation (Ref. [9],[10] and [11]) states that the above attributes are configurable as part of the firewall rules and each rule can configured to permit, drop and log.

The guidance material explains how to associate each rule with one or more specific network interfaces.

Testing	The evaluator shall perform the following tests: • The evaluator shall use the instructions in the operational guidance to configure baselines or anomaly-based rules for each attributes specified in IPS_ABD_EXT.1.1. The evaluator shall send traffic that does not match the baseline or matches the anomaly-based rule and verify the TOE applies the configured reaction. This shall be performed for each attribute in IPS_ABD_EXT.1.1. • Repeat the test assurance activity above to ensure that baselines or anomaly-based rules can be defined for each distinct network interface type supported by the TOE.

The evaluators configured custom attack signatures for each of the attributes specified in IPS_ABD_EXT.1.3. The evaluators configured both possible reactions (permit, deny) and enabled logging for each attack signature. The evaluators confirmed that the TOE identified traffic matching each signature and reacted as configured.

4.3.2 IPS_IPB_EXT.1.2 IPS_IPB_EXT.1 IP Blocking

TSS	The evaluator shall verify how good/bad lists affect the way in which traffic is analyzed with respect to processing packets. The TSS should also provide detail with the attributes that create a known good list, a known bad list, their associated rules, including how to define the source or destination IP address (e.g. a single IP address or a range of IP addresses). The evaluator shall also verify that the TSS identifies all the roles and level of access for each of those roles that have been specified in the requirement.

Good/bad address lists are constructed as part of IPv4/IPv6 policies and the TOE permits the use of the following header fields and data payloads for each supported protocol:

- IPv4: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; and IP Options.
- IPv6: Version; traffic class; flow label; payload length; next header; hop limit; source address; destination address; routing header; home address options.

The Security Administrator is able to perform the following functions:

- Modify the known-good and known-bad lists (of IP addresses or address ranges); and

- Configure the known-good and known-bad lists to override signature-based IPS policies.

Guidance	The evaluator shall verify that the administrative guidance provides instructions with how each role specified in the requirement can create, modify and delete the attributes of known good and known bad lists.

The provided guidance documentation (Ref. [9], [10] and [11]) states that the above attributes are configurable as part of the firewall rules and each rule can configured to permit, drop and log.

The guidance material explains how to associate each rule with one or more specific network interfaces.

Testing	The evaluator shall perform the following tests: • The evaluator shall use the instructions in the operational guidance to create a known-bad address list. Using a single IP address, a list of addresses or a range of addresses from that list, the evaluator shall attempt to send traffic through the TOE that would otherwise be allowed by the TOE and observe the TOE automatically drops that traffic. • The evaluator shall use the instructions in the operational guidance to create a known-good address list. Using a single IP address, a list of addresses or a range of addresses from that list, the evaluator shall attempt to send traffic that would otherwise be denied by the TOE and observe the TOE automatically allowing traffic. • The evaluator shall add conflicting IP addresses to each list and ensure that the TOE handles conflicting traffic in a manner consistent with the precedence in IPS_NTA_EXT.1.1.

The TOE configured security policies to both allow and deny traffic from certain subnets. The evaluators confirmed that, upon receiving traffic that matched these policies, the TOE reacted as configured (permitting or dropping the traffic).

The evaluators configured conflicting policies and confirmed that the TOE enforces policies in the administrator-defined order.

4.3.3 IPS_NTA_EXT.1 Network Traffic Analysis

4.3.3.1 IPS_NTA_EXT.1.1

TSS	The evaluator shall verify that the TSS explains the TOE's capability of analyzing IP traffic in terms of the TOE's policy hierarchy (precedence). The TSS should identify if the TOE's policy hierarchy order is configurable by the administrator for IPS policy elements (known-good lists, known-bad lists, signature-based rules, and anomaly-based rules). Regardless of whether the precedence is configurable, the evaluator shall verify that the TSS describes the default precedence as well as the IP analyzing functions supported by the TOE. The TSS associated with this requirement is assessed in the subsequent assurance activities.

Active security policies are enforced in a sequential order (based on the sequence number assigned to each policy). The administrator may change this sequence to suit their needs. If an IPS policy is attached to a security policy, it will be enforced at the same time as the security policy.

The default Implicit Deny policy will be enforced if the received traffic does not match any of the other active policies.

Guidance	If the precedence is configurable. The evaluator shall verify that the guidance explains how to configure the precedence.

Section 9 of the FortiOS handbook (Ref. [11]) contains a detailed breakdown of how the policy order of firewall objects is applied.

Testing	The testing associated with this requirement is assessed in the subsequent assurance activities.

N/A

4.3.3.2 IPS_NTA_EXT.1.2

TSS	The evaluator shall verify that the TSS indicates that the following protocols are supported: • IPv4; • IPv6; • ICMPv4; • ICMPv6; • TCP; and • UDP. The evaluator shall verify that the TSS describes how conformance with the identified protocols has been determined by the TOE developer. (e.g., third party interoperability testing, protocol compliance testing)

IPS rules can be associated with the following protocols:

- Internet Protocol (IPv4), RFC 791;
- Internet Protocol version 6 (IPv6), RFC 2460;
- Internet control message protocol version 4 (ICMPv4), RFC 792;
- Internet control message protocol version 6 (ICMPv6), RFC 2463;
- Transmission Control Protocol (TCP), RFC 793; and
- User Data Protocol (UDP), RFC 768.

The TOE's conformance with the protocols/RFCs listed above is determined during development. Compliance testing is performed as part of the development and release process with changes being made as required, ensuring conformance.

IPS policies can be applied to any TOE interface that can be included in an IPv4 or IPv6 policy. All interfaces capable of receiving network traffic can be utilised as a sensor, management or other type of interface as defined in the IPS EP

Guidance	The Guidance associated with this requirement is assessed in the subsequent assurance activities.

N/A

Testing	The testing associated with this requirement is addressed in the subsequent test assurance activities.

N/A

4.3.3.3 IPS_NTA_EXT.1.3

TSS	The evaluator shall verify that the TSS identifies all interface types capable of being deployed in the modes of promiscuous, and or inline mode as well as the interfaces necessary to facilitate each deployment mode (at a minimum, the interfaces need to support inline mode). The TSS should also provide descriptions how the management interface is distinct from sensor interfaces.

IPS policies can be applied to any TOE interface that can be included in an IPv4 or IPv6 policy. All interfaces capable of receiving network traffic can be utilised as a sensor (promiscuous or inline), management or other type of interface as defined in the IPS EP.

A 'management' interface, in the context of the IPS EP, can be considered any TOE interface configured for remote administration that does not have an IPS policy tied to it.

Guidance	The evaluator shall verify that the operational guidance provides instructions on how to deploy each of the deployment methods outlined in the TSS. The evaluator shall also verify that the operational guidance provides instructions of applying IPS policies to interfaces for each deployment mode. If the management interface is configurable the evaluator shall verify operational guidance explains how to configure the interface into a management interface. The evaluator shall verify that the operational guidance explains how the TOE sends commands to remote traffic filtering devices. Note: the secure channel configurations between the TOE and the remote device would be discussed as per FTP_ITC.1 (if the ST author selects other interface types) and/or FTP_TRP.1 (for interfaces in management mode) in the base PP.

The provided guidance documentation (Ref. [9], [10] and [11]) states that the above attributes are configurable as part of the firewall rules and each rule can configured to permit, drop and log.

Additionally the guidance material explains how to associate these rules with specific network interfaces.

Testing	The tests associated for this requirement have been completed in subsequent assurance activities in which promiscuous and inline interfaces are tested (e.g. tests for IPS_SBD_EXT.1.7) and in the requirement of FTP_ITC.1 (if the ST author selects other interface types) and/or FTP_TRP.1 (for interfaces in management mode) in the base PP.

N/A

4.3.4 IPS_SBD_EXT.1 Signature-Based IPS Functionality

4.3.4.1 IPS_SBD_EXT.1.1

TSS	The evaluator shall verify that the TSS describes what is comprised within a signature rule. The evaluator shall verify that each signature can be associated with a reaction specified in IPS_SBD_EXT.1.5. The evaluator shall verify that the TSS identifies all interface types capable of applying signatures and explains how rules are associated with distinct network interfaces. Where interfaces can be grouped into a common interface type (e.g., where the same internal logical path is used, perhaps where a common device driver is used) they can be treated collectively as a distinct network interface.

A signature rule is comprised of administrator-selected components which specify the parameters (traffic class, protocol, ports, etc.) that must be met to trigger a reaction (allow, drop, reset, etc.). Signature rules are authored in the following format:

F-SBID(--parameter1; --parameter2; --etc);

IPS policies can be applied to any TOE interface that can be included in an IPv4 or IPv6 policy. All interfaces capable of receiving network traffic can be utilised as a sensor (promiscuous or inline), management or other type of interface as defined in the IPS EP.

Guidance	The evaluator shall verify that the operational guidance provides instructions with how to create and/or configure rules using the following protocols and header inspection fields: • IPv4: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; and IP Options. • IPv6: Version; traffic class; flow label; payload length; next header; hop limit; source address; destination address; routing header; home address options. • ICMP: Type; Code; Header Checksum; and Rest of Header(varies based on the ICMP type and code). • ICMPv6: Type; Code; and Header Checksum. • TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options. • UDP: Source port; destination port; length; and UDP checksum. The evaluator shall verify that the operational guidance provides instructions with how to select and/or configure reactions specified in IPS_SBD_EXT.1.5 in the signature rules.

The provided guidance documentation (Ref. [9], [10] and [11]) states that the above attributes are configurable as part of the firewall rules and each rule can configured to permit, drop and log.

Additionally the guidance material explains how to associate these rules with specific network interfaces.

Testing	The evaluator shall perform the following tests: The evaluator shall use the instructions in the operational guidance to test that packet header signatures can be created and/or configured with the selected and/or configured reactions specified in IPS_SBD_EXT.1.5 for each of the attributes listed below. Each attribute shall be individually assigned to its own unique signature: • IPv4: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; and IP Options. • IPv6: Version; traffic class; flow label; payload length; next header; hop limit; source address; destination address; routing header; home address options. • ICMP: Type; Code; Header Checksum; and Rest of Header (varies based on the ICMP type and code). • ICMPv6: Type; Code; and Header Checksum;. • TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options. • UDP: Source port; destination port; length; and UDP checksum. • Using packet sniffers, the evaluator will generate traffic to trigger a signature and using packet captures will ensure that the reactions of each rule are performed as expected. Repeat the test assurance activity above to ensure that signature-based IPS policies can be defined for each distinct network interface type capable of applying signatures as supported by the TOE.

The evaluators configured custom attack signatures for each of the protocols and attributes listed above. The evaluators configured a variety of reactions (permit, deny) and enabled logging for each attack signature. The evaluators confirmed that the TOE identified traffic matching each signature and reacted as configured.

4.3.4.2 IPS_SBD_EXT.1.2

TSS	The evaluator shall verify that the TSS describes what is comprised within a string-based detection signature. The evaluator shall verify that each packet payload string-based detection signature can be associated with a reaction specified in IPS_SBD_EXT.1.5.

String-based detection signatures follow the format defined previously but will include a pattern field that allows administrators to define regular expressions for string matching.

Packet payload (string-based) detection signatures are placed within an IPS policy (as with other signature types) before being configured with the reactions specified earlier (allow, deny or reset).

Guidance	The evaluator shall verify that the operational guidance provides instructions with how to configure rules using the packet payload string-based detection fields defined in IPS_SBD_EXT.1.2. The operational guidance shall provide configuration instructions, if needed, to detect payload across multiple packets. The evaluator shall verify that the operational guidance provides instructions with how to configure reactions specified in IPS_SBD_EXT.1.5 for each string-based detection signature. The evaluator shall verify that the operational guidance provides instructions with how rules are associated with distinct network interfaces that are capable of being associated with signatures.

The provided guidance documentation (Ref. [9], [10] and [11]) indicates that the attributes defined in this requirement are configurable as part of the firewall rules and each rule can be configured to permit, drop and log.

Additionally, the guidance material explains how to associate each rule with one or more specific network interfaces.

Testing	The evaluator shall perform the following tests: Test 1: The evaluator shall use the instructions in the operational guidance to test that packet payload string-based detection rules can be assigned to the reactions specified in IPS_SBD_EXT.1.5 using the attributes specified in IPS_SBD_EXT.1.2. However it is not required (nor is it feasible) to test all possible strings of protocol data, the evaluator shall ensure that a selection of strings in the requirement is selected to be tested. At a minimum at least one string using each of the following attributes from IPS_SBD_EXT.1.2 should be tested for each protocol. The evaluator shall generate packets that match the string in the rule and observe the corresponding reaction is as configured. • Test at least one string of characters for ICMPv4 data: beyond the first 4 bytes of the ICMP header. • Test at least one string of characters for ICMPv6 data: beyond the first 4 bytes of the ICMP header. • TCP data (characters beyond the 20 byte TCP header): a) Test at least one FTP (file transfer) command: help, noop, stat, syst, user, abort, acct, allo, appe, cdup, cwd, dele, list, mkd, mode, nlst, pass, pasv, port, pass, quit, rein, rest, retr, rmd, rnfr, rnto, site, smnt, stor, stou, stru, and type. b) HTTP (web) commands and content: i. Test both GET and POST commands ii. Test at least one administrator-defined strings to match URLs/URIs, and web page content. c) Test at least one SMTP (email) state: start state, SMTP commands state, mail header state, mail body state, abort state. d) Test at least one string in any additional attribute type defined within [selection: [assignment: other types of TCP payload inspection]; • Test at least one string of UDP data: characters beyond the first 8 bytes of the UDP header; • Test at least one string for each additional attribute type defined in [assignment: other types of packet payload inspection]] The evaluator shall repeat one of the tests in Test 1 but generate multiple non-fragmented packets that contain the string in the rule defined. Repeat the test assurance activity above to ensure that signature-based IPS policies can be defined for each distinct network interface type capable of applying signatures as supported by the TOE.

The evaluators configured custom attack signatures for each of the protocols and characteristics listed above. The evaluators configured a variety of reactions (permit, deny) and enabled logging for each attack signature. The evaluators confirmed that the TOE identified traffic matching each signature and reacted as configured.

4.3.4.3 IPS_SBD_EXT.1.3

TSS	The evaluator shall verify that the TSS describes how the attacks defined in IPS_SBD_EXT.1.3 are processed by the TOE and what reaction is triggered when these attacks are identified.

Per Section 8.17 of the ST, packet payload (string-based) detection signatures must be placed within an IPS policy before being configured with the desired reaction(s).

The TOE is capable of detecting the following attack and scan types:

- IP Attacks
  - IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack); and
  - IP source address equal to the IP destination (Land attack).
- ICMP Attacks
  - Fragmented ICMP Traffic (e.g. Nuke attack); and
  - Large ICMP Traffic (Ping of Death attack).
- TCP Attacks
  - TCP NULL flags;
  - TCP SYN+FIN flags;
  - TCP FIN only flags; and
  - TCP SYN+RST flags.
- UDP Attacks
  - UDP Bomb Attack; and
  - UDP Chargen DoS Attack.
- Flooding a host (DoS attack)
  - ICMP flooding (Smurf attack, and ping flood);
  - TCP flooding (e.g. SYN flood); and
  - Flooding a network (DoS attack).
- Protocol and port scanning
  - IP protocol scanning;
  - TCP port scanning;
  - UDP port scanning; and
  - ICMP scanning.

These attacks are processed in the same manner as all other traffic passing through the TOE. The reaction is configurable by the administrator and can be set to Permit, Deny or Reset (for TCP).

Guidance	The evaluator shall verify that the operational guidance provides instructions with configuring rules to identify the attacks defined in IPS_SBD_EXT.1.3 as well as the reactions to these attacks as specified in IPS_SBD_EXT.1.5.

The provided guidance documentation (Ref. [9], [10] and [11]) states that the above attributes are configurable as part of the firewall rules and each rule can configured to permit, drop and log. The documentation also explains how to associate these rules with specific network interfaces.

Testing	The evaluator shall create and/or configure rules for each attack signature in IPS_SBD_EXT.1.3. For each attack, the TOE should apply its corresponding signature and enable it to each distinct network interface type capable of applying the signatures. The evaluator shall use packet captures to ensure that the attack traffic is detected by the TOE and a reaction specified in IPS_SBD_EXT.1.5 is triggered and stops the attack. Each attack should be performed one after another so as to ensure that its corresponding signature successfully identified and appropriately reacted to a particular attack.

The evaluators configured custom attack signatures for each of the attack signatures listed in IPS_SBD_EXT.1.3. The evaluators configured these signatures with each of the supported reactions (permit, deny or TCP reset) and enabled logging for each attack signature. The evaluators confirmed that the TOE identified traffic matching each signature and reacted as configured.

4.3.4.4 IPS_SBD_EXT.1.4

TSS	The evaluator shall verify that the TSS describes how the attacks defined in IPS_SBD_EXT.1.4 are processed by the TOE and what reaction is triggered when these attacks are identified.

Per Section 8.17 of the ST, packet payload (string-based) detection signatures must be placed within an IPS policy before being configured with the reactions supported by the TOE (allow, deny or reset). The TOE is capable of detecting the following attack and scan types:

- IP Attacks
  - IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack); and
  - IP source address equal to the IP destination (Land attack).
- ICMP Attacks
  - Fragmented ICMP Traffic (e.g. Nuke attack); and
  - Large ICMP Traffic (Ping of Death attack).
- TCP Attacks
  - TCP NULL flags;
  - TCP SYN+FIN flags;
  - TCP FIN only flags; and
  - TCP SYN+RST flags.
- UDP Attacks
  - UDP Bomb Attack; and
  - UDP Chargen DoS Attack.
- Flooding a host (DoS attack)
  - ICMP flooding (Smurf attack, and ping flood);
  - TCP flooding (e.g. SYN flood); and
  - Flooding a network (DoS attack).
- Protocol and port scanning
  - IP protocol scanning;
  - TCP port scanning;
  - UDP port scanning; and
  - ICMP scanning.

These attacks are processed in the same manner as all other traffic passing through the TOE. The reaction is configurable by the administrator and can be set to Permit, Deny or Reset (for TCP).

Guidance	The evaluator shall verify that the operational guidance provides instructions with configuring rules to identify the attacks defined in IPS_SBD_EXT.1.4 as well as the reactions to these attacks as specified in IPS_SBD_EXT.1.5.

The provided guidance documentation (Ref. [9], [10] and [11]) indicates that the above attacks specified in this requirement are configurable as part of the firewall rules and each rule can configured to permit, drop and/or log. The guidance material explains how to associate these rules with one or more specific network interfaces.

Testing	The evaluator shall configure individual signatures for each attack in IPS_SBD_EXT.1.4. For each attack, the TOE should apply its corresponding signature and enable it to each distinct network interface type capable of applying signatures. The evaluator shall use packet captures to ensure that the attack traffic is detected by the TOE and a reaction specified in IPS_SBD_EXT.1.5 is triggered and stops the attack. Each attack should be performed one after another so as to ensure that its corresponding signature successfully identified and appropriately reacted to a particular attack.

The evaluators configured custom attack signatures for each of the attack signatures listed in IPS_SBD_EXT.1.4. The evaluators configured each of the supported reactions (permit, deny or TCP reset) and enabled logging for each attack signature. The evaluators confirmed that the TOE identified traffic matching each signature and reacted as configured.

5 PROTECTION PROFILE SAR ASSURANCE ACTIVITIES

The following section addresses each of the assurance activities that correspond to the SARs claimed in the Security Target (Ref. [8]).

5.1 Development (ADV)

5.1.1 Basic functional specification (ADV_FSP.1)

Assurance activities	There are no specific assurance activities associated with these SARs. The functional specification documentation is provided to support the evaluation activities described in Section 4.2, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because the there is insufficient interface information, then an adequate functional specification has not been provided.

N/A

5.2 Guidance documentation (AGD)

5.2.1 Operational user guidance (AGD_OPE.1)

Assurance activities	Some of the contents of the operational guidance will be verified by the assurance activities in Section 4.2 and evaluation of the TOE according to the CEM. The following additional information is also required. The operational guidance shall at a minimum list the processes running (or that could run) on the TOE in its evaluated configuration during its operation that are capable of processing data received on the network interfaces (there are likely more than one of these, and this is not limited to the process that "listens" on the network interface). It is acceptable to list all processes running (or that could run) on the TOE in its evaluated configuration instead of attempting to determine just those that process the network data. For each process listed, the administrative guidance will contain a short (e.g., one- or two-line) description of the process' function, and the privilege with which the service runs. "Privilege" includes the hardware privilege level (e.g., ring 0, ring 1), any software privileges specifically associated with the process, and the privileges associated with the user role the process runs as or under.

The 'Parallel Path Processing - Life of a Packet' section of the FortiOS Handbook (Ref. [11]) provides the following list of processes that affect packets within the device as well as a high level description of each one.

- Ingress packet flow
  - Network Interface
  - TCP/IP stack
  - DoS ACL
  - DoS Policy
  - IP integrity header checking
  - IPsec VPN decryption
- Admission Control

    - Quarantine
    - FortiTelemetry
    - User Authentication
- Kernel
    - Destination NAT
    - Routing
    - Stateful inspection/Policy
    - Lookup/Session management
    - Session Helpers
    - User Authentication
    - Device Identification
    - SSL VPN
    - Local Management Traffic
- UTM/NGFW
    - Flow-based inspection
    - NTurbo
    - IPSA
    - Proxy-based inspection
- Kernel
    - Forwarding
    - Source NAT (SNAT)
- Egress packet flow
    - IPsec VPN Encryption
    - Botnet check
    - Traffic shaping
    - WAN Optimization
    - TCP/IP stack
    - Network Interface

Assurance activities	The operational guidance shall contain instructions for configuring the 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.

No configuration of the cryptographic engine is required by TOE users, therefore no guidance pertinent to this requirement is provided.

Assurance activities	The documentation must describe the process for verifying updates to the TOE, either by checking the hash or by verifying a digital signature. The evaluator shall verify that this process includes the following steps: 1. For hashes, a description of where the hash for a given update can be obtained. For digital signatures, instructions for obtaining the certificate that will be used by the FCS_COP.1(2) mechanism to ensure that a signed update has been received from the certificate owner. This may be supplied with the product initially, or may be obtained by some other means. 2. 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). 3. Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature.

The FortiOS Handbook (Ref. [11]) provides TOE administrators with information on the upgrade process for the TOE, including the mechanisms used to verify updates, how updates are obtained and how to initiate the upgrade process.

Assurance activities	The TOE will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.

The Fortinet CC/FIPS guide (Ref. [9]) identifies the CC documents to which the TOE is conformant (PP/EPs) and the functionality provided by the TOE which was included in the scope of the evaluation.

5.2.2 Preparative procedures (AGD_PRE.1)

Assurance activities	Preparative procedures must include a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target). The documentation should be in an informal style and should be written with sufficient detail and explanation that they can be understood and used by the target audience (which will typically include IT staff who have general IT experience but not necessarily experience with the TOE product itself). Preparative procedures must be provided for every Operational Environment that the product supports as claimed in the Security Target and must adequately address all platforms claimed for the TOE in the Security Target. The preparative procedures must include a) instructions to successfully install the TSF in each Operational Environment; b) instructions to manage the security of the TSF as a product and as a component of the larger operational environment; and c) instructions to provide a protected administrative capability.

The 'FIPS 140-2 and CC Compliant Operation for FortiOS 5.4' guide (Ref. [9]) and the FortiOS Handbook (Ref.[11]) provide the instructions need to install and manage the TOE in accordance with the FWcPP, IPSEP and VPNEP. These instructions apply to all given platforms and are of sufficient detail to allow the target audience to understand and use the TOE.

5.3 Lifecycle support (ALC)

5.3.1 Labelling of the TOE (ALC_CMC.1)

Assurance activities	The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. The evaluator shall ensure that this identifier is sufficient for an acquisition entity to use in procuring the TOE (including the appropriate administrative guidance) as specified in the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

The evaluators verified that the TOE reference used is consistent across the TOE and associated documentation.

5.3.2 TOE CM coverage (ALC_CMS.1)

Assurance activities	The 'evaluation evidence required by the SARs' in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the assurance activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component.

The evaluators verified that ALC_CMC.1 is satisfied, hence ALC_CMS.1 is also satisfied.

5.4 Testing (ATE)

5.4.1 Independent testing – conformance (ATE_IND.1)

Assurance activities	The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the CEM and the body of this PP's Assurance Activities. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluator must document in the test plan that each applicable testing requirement in the ST is covered. The test plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary. The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by this PP and used by the cryptographic protocols being evaluated (IPsec, TLS/HTTPS, SSH). The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful rerun of the test, the report would show a "fail" and "pass" result (and the supporting details), and not just the "pass" result.

The evaluators developed a detailed test report for each PP/EP (EFS-T045-FWcPP-TR, EFS-T045-VPN-TR and EFS-T045-IPS-TR)to address all aspects of this requirement. The TRs discuss the test configuration, test cases, expected results and actual results.

A sampling strategy was used for testing – this strategy was agreed upon by the Lab, ASD and NIAP.

5.5 Vulnerability assessment (AVA)

5.5.1 Vulnerability survey (AVA_VAN.1)

Assurance activities	As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in network infrastructure devices and the implemented communication protocols in general, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, a test would be suitable at the assurance level of this PP. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.

The vulnerability analysis performed by the evaluators is detailed in the NDPP Test Report (EFS-T045-FWcPP-TR). The vulnerability analysis included the following tests.

- Testing the TOEs response to Malformed Packets using undefined RFC variables
- Testing the TOEs response to Malformed Packets using junk variables;
- Testing whether the form fields provided by the TOE are susceptible to buffer overflow vulnerabilities;
- Testing whether the form fields provided by the TOE are susceptible to cross-site scripting attacks (XSS); and
- Testing whether the form fields provided by the TOE are susceptible to input validation vulnerabilities.

---- END OF DOCUMENT ----