Security Target

FortiGate™ UTM appliances running FortiOS™ 5.0 Patch Release 10

Common Criteria Evaluation with
Network Device Protection Profile v1.1
Stateful Traffic Filter Firewall Extended Package v1.0

Document Version: 1.7
Date: April 13, 2015

Prepared For:

Fortinet, Inc

326 Moodie Drive
Ottawa, ON K2H 8G3, Canada
www.fortinet.com

Prepared By:

CGI Global IT Security Labs.

1410 Blair Place, 7th floor
Ottawa, ON K1J 9B9, Canada
www.cgi.com/securitylab

Revision History

Ver #	Description of changes	Modified by	Date
0.1	Initial draft to lab	Shawn Pinet	June 26, 2013
0.2	Additional details on the 2 crypto modules, additional platform details.	Shawn Pinet	July 4, 2013
0.3	Updated entropy source details, response to lab EAP questions	Shawn Pinet	August 27, 2013
0.4	Updates to address Lab Acceptance	Shawn Pinet	August 30, 2013
0.5	Updates to address lab ASE OR#1 and ASE OR#2	Shawn Pinet	September 27, 2013
0.6	Updates to address remaining minor ASE observation	Shawn Pinet	September 30, 2013
0.7	Updates from Fortinet for additional supported HTTPS/TLS ciphers	Shawn Pinet	November 6, 2013
0.8	Updates to the version number for consistency	Shawn Pinet	November 12, 2013
0.9	Updates to respond to lab and CSEC observations	Shawn Pinet	January 3, 2014
1.0	Updated hardware models	Shawn Pinet	March 3, 2014
1.1	Updates adding FAZ over TLS, remove IPSEC	Shawn Pinet	March 26, 2014
1.2	Updates to address testing observations	Shawn Pinet	April 8, 2014
1.3	Updates to respond to lab error states	Danielle Freebourne	October 9, 2014
1.4	Finalization of document numbers, build numbers	Danielle Freebourne	March 4, 2015
1.5	Updates for CAVP Certs for FortiOS	Danielle Freebourne	March 17, 2015
1.6	Update conformance claim, public documentation references	Danielle Freebourne	March 24, 2015
1.7	Updates to the crypto modules to provide additional information requested from the CB	Danielle Freebourne	April 13, 2015

TABLE OF CONTENTS

1 Introduction 7
1.1 ST Reference 7
1.2 Target of Evaluation Reference 7
1.3 Conventions 7
1.4 TOE Overview 8
1.5 TOE Description 8
1.5.1 Physical Boundary 8
1.5.2 Logical Boundary 11
1.5.3 Hardware, firmware, and Software Supplied by the IT Environment 15
1.5.4 Product Physical/Logical Features and Functions not included in the TOE Evaluation 15

2 Conformance Claims 17
2.1 Common Criteria Conformance Claim 17
2.2 Protection Profile Conformance Claim 17

3 Security Problem Definition 18
3.1 Threats 18
3.2 Organizational Security Policies 19
3.3 Assumptions 19

4 Security Objectives 20
4.1 Security Objectives for the TOE 20
4.2 Security Objectives for the Operational Environment 20

5 Extended Security Requirement Components Definition 22
5.1 Extended TOE Security Functional Requirement Components 22
5.1.1 FAU_STG_EXT.1 External Audit Trail Storage 22
5.1.2 FCS_CKM_EXT.4 Cryptographic Key Zeroization 22
5.1.3 FCS_HTTPS_EXT.1 Extended: HTTPS 23
5.1.4 FCS_TLS_EXT.1 Extended: TLS 23
5.1.5 FCS_RBG_EXT.1 Extended: Random Bit Generation 24
5.1.6 FIA_PMG_EXT.1 Password Management 25
5.1.7 FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism 25
5.1.8 FIA_UIA_EXT.1 Extended: Password-based Authentication and Identification Mechanism 26
5.1.9 FPT_APW_EXT.1 Extended: Protection of Administrator Passwords 26
5.1.10 FPT_SKP_EXT.1 Extended: Protection of TSF data (for reading of all symmetric keys) 27
5.1.11 FPT_TST_EXT.1 Extended: TSF testing 27
5.1.12 FPT_TUD_EXT.1 Extended: Management of TSF Data 27
5.1.13 FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking 28
5.1.14 FFW_RUL_EXT.1 Stateful Traffic Filtering 28
5.2 Extended TOE Security Assurance Requirement Components 32

6 Security Requirements 33
6.1 Security Functional Requirements 33
6.1.1 Security Audit (FAU) 34
6.1.2 Cryptographic Support (FCS) 36
6.1.3 User Data Protection (FDP) 37
6.1.4 Identification and Authentication (FIA) 38
6.1.5 Security Management (FMT) 38
6.1.6 Protection of the TSF (FPT) 39
6.1.7 TOE Access (FTA) 39
6.1.8 Trusted Path/Channels (FTP) 40

6.1.9 Stateful Traffic Filtering (FIA) 41
6.2 Security Assurance Requirements *43*

7 TOE Summary Specification 45
7.1 Security Audit *45*
7.2 Cryptographic Support *46*
7.2.1 Entropy Source and Random Bit Generation 49
7.2.2 Cryptographically Trusted Paths 50
7.2.3 HTTPS 50
7.2.4 TLS 50
7.2.5 Cryptographic Self Tests and TOE Update Integrity 51
7.2.6 Conformance to NIST SP800-56 52
7.2.7 Key and CSP storage and zeroization 52
7.3 User Data Protection *52*
7.4 Identification and Authentication *53*
7.4.1 Web/HTTPS 53
7.4.2 Local Serial Console 53
7.4.3 Universal Serial Bus 54
7.5 Security Management *54*
7.5.1 Local Console CLI 54
7.5.2 Web UI 54
7.6 Protection of the TSF *55*
7.6.1 Cryptographic Key and Password Storage 55
7.6.2 FortiGate™ Product Updates 55
7.6.3 Self-Tests 56
7.7 TOE Access *56*
7.8 Trusted Path/Channels *57*
7.9 Stateful Traffic Firewall *57*
7.9.1 Overview of Firewall Functionality 57
7.9.2 Firewall Interoperability 58
7.9.3 Overview of Startup Process 58
7.9.4 Firewall Error Modes 59
7.9.5 Default Stateful Traffic Filtering 59

8 Rationale 61
8.1 Dependency Rationale *61*

9 Acronyms 65

10 Appendix A – Hardware Platform Details 66
10.1 Desktop Form Factor *66*
10.2 1U Form Factor *66*
10.3 2U Form Factor *67*
10.4 3U Form Factor *67*
10.5 Blade Form Factor *68*

LIST OF TABLES

Table 1 – Threats 18
Table 2 – Organizational Security Policies 19
Table 3 – Assumptions 19
Table 4 – TOE Security Objectives 20
Table 5 – Operational Environment Security Objectives 21
Table 6 – TOE Security Functional Requirements 33
Table 7 – Auditable Events 34
Table 8 – Security Assurance Requirements 43
Table 9 – Software Cryptographic Module Algorithms 46
Table 10 – FortiOS Hardware Cryptographic Certificates 48
Table 11 –Functional Requirements Dependencies 61
Table 12 – Acronyms 65

LIST OF FIGURES

Figure 1 – TOE Physical Boundary 9

1 INTRODUCTION

This section identifies the Security Target (ST), Target of Evaluation (TOE), document conventions, and terminology. It also provides TOE overview and describes the hardware and software that make up the TOE as well as the physical and logical boundaries of the TOE.

1.1 ST Reference

ST Title	FortiGate™ UTM appliances running FortiOS™ 5.0 Patch Release 10
ST Revision	1.7
ST Publication Date	April 13, 2015
ST Author	CGI Global IT Security Labs
	Shawn Pinet – Senior Consultant
	Danielle Freebourne - Consultant

1.2 Target of Evaluation Reference

TOE Developer	Fortinet, Inc
TOE Name	FortiGate™ UTM appliances running FortiOS™ 5.0 Patch Release 10
TOE Version	5.0 Patch Release 10 FIPS/CC build 305 with FTR-ENT1

1.3 Conventions

The Common Criteria allows for assignment, refinement, selection and iteration operations to be performed on security functional requirements. All of these operations are used within this ST. These operations are performed as described in Part 2 of the CC, and selected presentation choices are discussed below to aid the Security Target reader:

- An assignment operation is indicated by [*italicized text within brackets*].
- Selections are denoted by [underlined text within brackets].
- Refinement of security requirements is identified using **bold text**. Any text removed is indicated with a strikethrough (Example: ~~TSF~~).
- Iterations are identified by appending a number in parentheses following the component title, for example, FIA_UAU.1 (1) and FIA_UAU.1 (2) refer to two iterations of the FIA_UAU.1 security functional requirement component.

1.4 TOE Overview

The TOE is the referenced network appliances as detailed in section 1.5.1 running version 5.0 Patch Release 10 of the FortiOS code in stand-alone mode. The TOE is designed to provide next-generation firewall services ensuring network protection for Internet Protocol version 4 (IPv4) and Internet Protocol version 6 (IPv6) networks. The TOE is capable of robust filtering based on information contained in IPv4, IPv6, ICMPv4, ICMPv6, TCP and UDP headers as specified by their respective RFC's. Additionally the TOE is capable of content inspection of FTP and H.323 protocols to work with the dynamic nature of these protocols.

The TOE has extensive logging capabilities, as described by section 6.1.1 of this document. These include, but are not limited to the firewall rules described above, administrative actions and logging and tampering or misuse of the trusted cryptographic channels. These audit logs are capable of being exported to an external FortiAnalyzer™ audit server over a cryptographically protected channel for further analysis and inspection.

The TOE implements NIST approved cryptography, validated through numerous FIPS and CAVP validations detailed in Table 9 and Table 10 of this document. This cryptography is used to secure communications to trusted administrators, remote authentication sources and to secure generated audit logs in transit to the FortiAnalyzer server for additional inspection. User administration sessions are capable over the local console or secured over HTTPS to a web based GUI using validated cryptography. To ensure proper random number generation capable of generating keys with 256 bits of strength the TOE has been equipped with a dedicated hardware noise source which provides entropy collected from the ambient environment in which the product operates. This noise source is continually monitored for its ongoing health and proper operation.

The TOE also offers the ability to verify through cryptographic signatures that product updates are valid, and will reject any updates without the appropriate Fortinet signature. The TOE will ensure during boot up that the health of the TOE has not been compromised through a variety of checks. These checks include health testing of the entropy source to ensure that the ambient environment is producing sufficient entropy for the seeding of the cryptographic module. The TOE also implements FIPS 140-2 level 2 hardware requirements for potential tampering and the firmware is inspected on startup as described in the FIPS 140-2 level 1 firmware integrity checks.

1.5 TOE Description

This section primarily addresses the physical and logical components of the TOE included in the evaluation.

1.5.1 Physical Boundary

The physical scope of the TOE includes the TOE hardware and firmware as well as a FTR-ENT1 to provide the hardware noise source. Details on the environment and TSFI's are shown below:

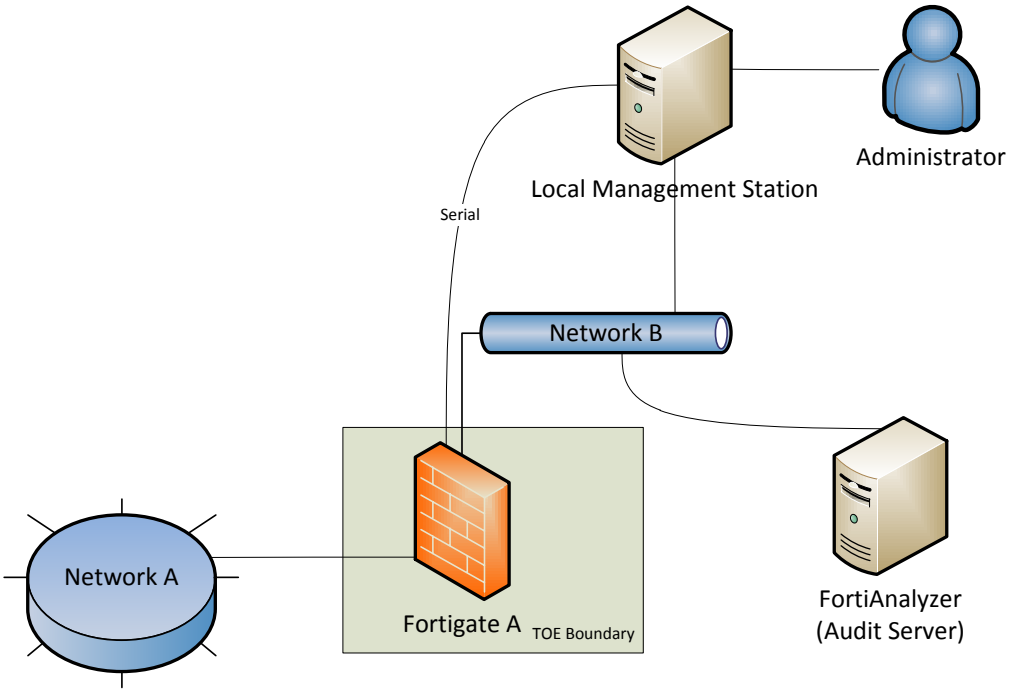

Figure 1 – TOE Physical Boundary

The following TOE hardware platforms are claimed for this evaluation. Each of these hardware platforms requires a FTR-ENT1 to seed the TOE cryptographic system with entropy from the ambient environment.

1.5.1.1 Desktop Hardware Models

- FortiGate-20C
- FortiWiFi-20C
- FortiGate-30D
- FortiWiFi-30D
- FortiWiFi-30D-PoE
- FortiGate-40C
- FortiWiFi-40C
- FortiGate-60C
- FortiGate-60D
- FortiWiFi-60C
- FortiGate-60D-PoE
- FortiWiFi-60D
- FortiGate-80C
- FortiWiFi-80CM
- FortiGate-90D
- FortiGate-90D-PoE
- FortiGate-110C
- FortiGate-111C

1.5.1.2 1U Hardware Models

- FortiGate-100D
- FortiGate-140D
- FortiGate-140D-PoE
- FortiGate-200B
- FortiGate-200B-PoE
- FortiGate-200D
- FortiGate-240D
- FortiGate-300C
- FortiGate-300D
- FortiGate-310B
- FortiGate-311B
- FortiGate-500D
- FortiGate-600C
- FortiGate-620B
- FortiGate-621B
- FortiGate-800C

1.5.1.3 2U Hardware Models

-
- FortiGate-1000C
- FortiGate-1000D
- FortiGate-1200D
- FortiGate-1240B
- FortiGate-1500D
- FortiGate-280D-PoE
- FortiGate-3040B
- FortiGate-3140B
- FortiGate-3240C

1.5.1.4 3U Hardware Models

- FortiGate-3600C
- FortiGate-3700D
- FortiGate-3950B
- FortiGate-3951B

1.5.1.5 FortiGate 5000 series Hardware Models

The FortiGate 5000 series chassis are modular enclosures for blade systems. The following blade systems are capable of running in the evaluated configuration:

- FortiGate-5020 (2 Blade Slots)
- FortiGate-5060 (6 Blade Slots)
- FortiGate-5140B (14 Blade Slots)

These FortiGate series chassis require one or more of the following hardware blades:

- FortiGate-5001A
- FortiGate-5001B
- FortiGate-5001C
- FortiGate-5001D
- FortiGate-5101C
- FortiSwitch-5203B

1.5.1.6 Guidance Documentation

The following lists the TOE Guidance Documentation[1] to install, configure, and maintain the TOE.

FortiOS™ Handbook for FortiOS 5.0 01-5010-99686-20150219 February 19, 2015

FortiGate™ Log Message Reference v5.0 Patch Release 10 01-510-112804-20150313 March 13, 2015

FortiOS™ CLI Reference for FortiOS 5.0 01-509-99686-20150226 February 26, 2015

FIPS 140-2 and Common Criteria Compliant Operation for FortiOS™ 5.0.10 01-510-267768-20150206 March 20, 2015

FortiAnalyzer v5.0 Patch Release 9 Administration Guide 05-509-187572-20141020 October 20, 2014

FortiGate Appliances with FortiOS 5.0 (NDPP Compliant) Product Architectural Description 0.3 February 6, 2015

1.5.2 Logical Boundary

This section outlines the boundaries of the security functionality of the TOE; the logical boundary of the TOE includes the security functionality described in the following sections.

1.5.2.1 Security Audit

The TOE is capable of generating and securely transmitting Security Audit logs to a remote, trusted FortiAnalyzer server for further processing and review. The TOE will generate auditable events as specified in the NDPP which may help indicate a number of potential security concerns including resonance, password guessing and tampering with the trusted paths and channels. For all auditable events the TOE will associate a user (either IP address or with administrative credentials) to the session and use this identifier for all logging to the audit server.

The TOE can generate audit logs for a variety of security events. These include basic events such as hits against firewall rules and will include information which is tracked by the TOE and exported for later analysis and review via a trusted channel. This information includes information such as source, destination, port and protocol as required by the Firewall EP.

In addition to generating audit records, the TOE monitors auditable events and provides an administrator-configurable pattern threshold for determining interesting traffic on the firewall which may be an indication of a potential security violation. Once the TOE has detected a potential security

[1] Guidance documents are publicly available and listed on the Fortinet documentation website at http://docs.fortinet.com/

violation, an alarm message can be displayed at the TOE's local console as well as at each active remote administrative session. The event will also be logged to the remote FortiAnalyzer audit server. An authorized administrator may delete the local audit trail. An authorized administrator may configure these or additional auditable events, back-up audit data to an external source and manage audit data storage.

The auditing function is supported by reliable timestamps provided by the TOE.

1.5.2.2 Cryptographic Support

The TOE's cryptographic modules are FIPS PUB 140-2 validated and meet Security Level 1 overall and Security Level 2 for cryptographic module ports and interfaces, roles, services and authentication, and design assurance. The TOE is capable of generating cryptographic keys using a NIST SP 800-90B compliant random bit generator seeded with a minimum of 256 bits of entropy by the dedicated hardware based noise source. These keys are created, managed and destroyed to provide cryptographic services to the network. The TOE is also capable of importing cryptographic keys and certificates from outside the TOE boundary. Cryptographic keys as well as other CSPs[2] are zeroized by the FIPS compliant modules when no longer required and the TOE offers a function to zeroize this data on demand.

The TOE is designed such that the cryptographic keys and other CSPs are not exposed through the various interfaces made available to the TOE administrator(s). Passwords including administrative passwords and pre-shared keys are stored on the TOE in the configuration file. These passwords are stored in encrypted format by the TOE using an AES-128 generated by the TOE when initialized to obscure the credentials. Credentials entered on one of the administrative consoles are encrypted with this key and compared to the encrypted format stored in the configuration file. Certificates are not viewable from any interface and may only be imported to the TOE through the HTTPS GUI which is a cryptographically protected and trusted channel.

The TOE implements a HTTPS GUI and has compatibility with a wide variety of other products. More information on vendor compatibility, integration and known issues can be located on the Fortinet knowledgebase (KB[3]). Additionally the TOE protects communications with FortiAnalyzer via TLS.

1.5.2.3 User Data Protection

The TOE ensures that all information is zeroized on allocation of memory to ensure that all memory is cleared of residual information prior to being written to.

1.5.2.4 Identification and Authentication

All administration requires authentication by user identification and password mechanism. Administration may either be performed locally using the Local Console CLI or remotely using the Network Web-Based GUI. When authenticating to the TOE it supports complex configurable password rules and supports complex character sets.

[2] Critical Security Parameters
[3] The Fortinet KB is at http://kb.fortinet.com/

Any individual attempting to log on for an interactive session will be shown a warning message that they must accept prior to being presented with a prompt to attempt their authentication.

1.5.2.5 Security Management

The TOE provides remote and local administrative interfaces that permit the administrative roles to configure and manage the TOE. In the evaluated configuration the TOE is connected to two or more networks and remote administration request data flows from a Network Management Station to the TOE. In each configuration there is also a Local Console, located within the physically secured area described within the NDPP and consists of a physical serial[4] interface to the TOE.

An administrator account is associated with an access profile, which determines the permissions of the individual administrator. Additionally, each FortiGate™ unit comes with a default administrator account with all permissions, which may not be deleted but may have its credentials changed. The term 'authorized administrator' is used throughout this ST to describe an administrator given the appropriate permission to perform tasks as required.

These tasks include, but are not limited to configuring appropriate cryptographic protocols available for negotiation, the capacity to query the version information and the ability to update the TOE to a new version.

1.5.2.6 Protection of the TSF

Inter-TSF communications are protected to ensure availability, confidentiality and detection of modification. This is accomplished through the usage of cryptographic communications for any and all communications with remote IT entities, other components of the TOE and remote administrators. By default detection of modification and audit logging is enabled on TLS connections.

The TOE prevents the reading of all administrator passwords, pre-shared keys, symmetric keys and private keys through encrypting them with AES-128 prior to storing them into the TOE configuration file. These keys are viewable through the TSFI's only as this encrypted value and the value will be shown when a full configuration is shown or backed up by the administrator. Certificates cannot be viewed through any interface once loaded into the TOE.

The TOE is capable of querying its current version and displaying it back to the administrator via the trusted interfaces. The TOE also provides a method to verify and update product updates through querying the version of the TOE through any of the administrative interfaces. Updates to the TOE software is verified by the TOE during the initial phase of the update process. During this process the TOE verifies the candidate update is signed by the developer's 2048 bit RSA key in order to ensure the authenticity of the update. This cryptographic key is used for all FIPS firmware images.

The TOE maintains its own timestamp which are free from outside interference. This timestamp is used for the purposes of generating audit logs and other time-sensitive operations on the TOE including cryptographic key regeneration intervals.

[4] Some models implement this serial interface as a Universal Serial Bus (USB) interface and requires the FortiExplorer software client to be installed on the Administrator's workstation.

The TOE implements a number of self-tests on start-up to ensure the correct operation and configuration of the TOE. These include but are not limited to hardware and entropy source self-tests, checksums of the binaries and operation of the FIPS approved cryptographic module. Additionally the TOE maintains ongoing health tests associated with the FIPS cryptographic module and the hardware noise source.

1.5.2.7 TOE Access

The TOE is capable of terminating both local and remote administrative sessions upon detection of administrator inactivity. The TOE is also capable of terminating a remote session upon request from a remote administrator.

The TOE provides administrators with a configurable warning banner prior to initiating any interactive session with the administrator.

1.5.2.8 Trusted Path/Channels

A cryptographically protected trusted communications channel is required for all communications with the external remote authentication server and with the FortiAnalyzer audit server. For the purposes of auditing the TOE is capable of securing its audit server communications via TLS. The usage of this protocol ensures that the TOE will protect the credentials contained in the authentication request from disclosure and raise an audit log entry should the TOE detect modification in transit. The TOE or the remote peer may initiate this cryptographically protected channel.

The TOE will ensure that cryptographically protected sessions to the HTTPS GUI are used to establish a trusted path between the TOE and the trusted remote administrator. This path will be used for both the initial administrator authentication and all remote administration requests and is terminated upon session timeout or explicit request from the administrator.

1.5.2.9 Stateful Traffic Filtering

The TOE implements a stateful firewall which is compliant with the NDPP EP for stateful firewall inspection. 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. 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.

The TOE can create firewall rules based on a number of security attributes located in the header information of traffic arriving on a specific interface. Rules can be created based on a number of traffic protocols including the RFC's for ICMPv4, ICMPv6, IPv4, IPv6, TCP and UDP. Attributes of these protocols such as IP address, transport protocol, type, code and port can be used to provide more granular access control policies. The TOE also supports advanced protocols including FTP and H.323

which have non-static ports during their negotiation. The TOE is capable of inspecting this traffic to understand what is expected during these information flows.

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 an 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.

1.5.3 Hardware, firmware, and Software Supplied by the IT Environment

The following hardware, firmware and software, which are supplied by the IT environment, are excluded from the TOE boundary.

- Local management including
  - Local Console Software (FortiExplorer™ 2.2 or above and a Serial Console client)
  - Web Browser
- Logging Server
  - FortiAnalyzer appliance

1.5.4 Product Physical/Logical Features and Functions not included in the TOE Evaluation

The FortiGate™ appliances are capable of a variety of functions and configurations which are not covered by the NDPP. The TOE is capable of this functionality however the following features have not been examined as part of this evaluation:

- High-Availability
- FortiExplorer client
- Anti-spam
- Content filtering
- Web filtering
- IPSEC and SSL VPN gateway functionality
- Antivirus
- NAT
- Intrusion detection/prevention
- SSH
- Use of syslog
- FortiToken and FortiSSO Authentication
- Stream Control Transmission Protocol (SCTP), BGP, RIP, NTP and DHCP protocols
- Usage of the boot-time configuration menu to upgrade the TOE[5]

[5] This menu presents itself during a reboot of the TOE and contains options for zeroization of the TOE and for a factory reset of the TOE to a specified firmware version. The zeroization of the TOE is in scope but the upgrade shall be performed through the web UI.

2 CONFORMANCE CLAIMS

2.1 Common Criteria Conformance Claim

The Security Target is conformant to Common Criteria Version 3.1 Revision 4, September 2012, Part 2 extended and Part 3 conformant.

2.2 Protection Profile Conformance Claim

The Security Target is conformant to the:

- Network Devices Protection Profile (NDPP) v1.1, June 8, 2013, including the following optional requirements [TLS and TLS/HTTPS].
- Network Devices Protection Profile Errata #2, 14 January 2014
- The NDPP Extended Package Stateful Traffic Filter Firewall v1.0, December 19, 2011.

3 SECURITY PROBLEM DEFINITION

This section defines the security problem which the TOE and its operational environment are supposed to address. Specifically, the security problem makes up the following:

- Any known or assumed threats countered by the TOE or its operational environment.
- Any organizational security policies with which the TOE must comply.
- Any assumptions about the security aspects of the environment and/or of the manner in which the TOE is intended to be used.

This section identifies assumptions as A.*assumption*, threats as T.*threat* and policies as P.*policy*.

3.1 Threats

This section identifies the threats to the assets against which protection is required by the TOE or by the security environment.

The table below lists threats applicable to the TOE and its operational environment:

Table 1 – Threats

Threat	Description
T.ADMIN_ERROR	An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNDETECTED_ACTIONS	Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.
T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T.USER_DATA_REUSE	User data may be inadvertently sent to a destination not intended by the original sender.
T.NETWORK_DISCLOSURE	Sensitive information on a protected network might be disclosed resulting from ingress- or egress-based actions.
T.NETWORK_ACCESS	Unauthorized access may be achieved to services on a protected network from outside that network, or alternately services outside a protected network from inside the protected network.
T.NETWORK_MISUSE	Access to services made available by a protected network might be used counter to Operational Environment policies.
T.NETWORK_DOS	Attacks against services inside a protected network, or indirectly by virtue of access to malicious agents from within a protected network, might lead to denial of services otherwise available within a protected network.

3.2 Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. The following table lists Organizational Security Policies (OSP) applicable to the TOE and its operational environment:

Table 2 – Organizational Security Policies

OSP	Description
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

3.3 Assumptions

This section describes the security aspects of the environment in which the TOE is intended to operate. The following specific conditions are assumed to exist in an environment where the TOE is employed.

Table 3 – Assumptions

Assumption	Description
A.NO_GENERAL_PURPOSE	It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
A.CONNECTIONS	It is assumed that the TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.

4 SECURITY OBJECTIVES

Security objectives are concise, abstract statements of the intended solution to the problem defined by the security problem definition. This high-level solution is divided into two parts: the security objectives for the TOE, and the security objectives for the TOE's operational environment. This section identifies the security objectives for the TOE and its supporting environment.

4.1 Security Objectives for the TOE

The IT security objectives for the TOE are as follows:

Table 4 – TOE Security Objectives

Security Objective	Description
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
O.VERIFIABLE_UPDATES	The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the Administrator to be unaltered and (optionally) from a trusted source.
O.SYSTEM_MONITORING	The TOE will provide the capability to generate audit data and send those data to an external IT entity.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.TOE_ADMINISTRATION	The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.
O.RESIDUAL_INFORMATION_CLEARING	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.SESSION_LOCK	The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.
O.ADDRESS_FILTERING	The TOE will provide the means to filter and log network packets based on source and destination addresses.
O.PORT_FILTERING	The TOE will provide the means to filter and log network packets based on source and destination transport layer ports.
O.STATEFUL_INSPECTION	The TOE will determine if a network packet belongs to an allowed established connection before applying the ruleset.
O.RELATED_CONNECTION_FILTERING	For specific protocols, the TOE will dynamically permit a network packet flow in response to a connection permitted by the ruleset.

4.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are addressed below:

Table 5 – Operational Environment Security Objectives

Security Objective	Description
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.CONNECTIONS	TOE administrators will ensure that the TOE is installed in a manner that will allow the TOE to effectively enforce its policies on network traffic flowing among attached networks.

5 EXTENDED SECURITY REQUIREMENT COMPONENTS DEFINITION

This section defines the extended Security Functional Requirements (SFRs) and extended Security Functional Assurance Requirements (SARs) met by the TOE. All the extended components have been drawn from the Network Device Protection Profile (NDPP) v1.1 and the NDPP Extended Package Stateful Traffic Filter Firewall v1.0.

5.1 Extended TOE Security Functional Requirement Components

This section specifies the extended SFRs for the TOE.

5.1.1 FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.1 External Audit Trail Storage requires the TSF to use an external IT entity for audit data storage. It is modeled after FAU_STG.1, and is considered to be part of the FAU_STG family.

Management: FAU_STG_EXT.1

There are no management activities foreseen.

Audit: FAU_STG_EXT.1

There are no auditable events foreseen.

FAU_STG_EXT.1 External Audit Trail Storage

Hierarchical to:	No other components
Dependencies:	FAU_GEN.1 Audit data generation FTP_ITC.1 Inter-TSF trusted channel
FAU_STG_EXT.1.1	The TSF shall be able to [selection: transmit the generated audit data to an external IT entity, receive and store audit data from an external IT entity] using a trusted channel implementing the [selection: IPsec, SSH, TLS, TLS/HTTPS] protocol.

5.1.2 FCS_CKM_EXT.4 Cryptographic Key Zeroization

FCS_CKM_EXT.4 Cryptographic key zeroization requires cryptographic keys and cryptographic critical security parameters to be zeroized. It is modeled after FCS_CKM.4, and is considered to be part of the FCS_CKM family.

Management: FCS_CKM_EXT.4

There are no management activities foreseen.

Audit: FCS_CKM_EXT.4

There are no auditable events foreseen.

FCS_CKM_EXT.4 Cryptographic Key Zeroization

Hierarchical to:	No other components
Dependencies:	FDP_ITC.1 Import of user data without security attributes, or

	FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation
FCS_CKM_EXT.4.1	The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

5.1.3 FCS_HTTPS_EXT.1 Extended: HTTPS

FCS_HTTPS_EXT.1 Extended: HTTPS requires that HTTPS be implemented. It belongs to a new family defined for the FCS Class.

Management: FCS_HTTPS_EXT.1

There are no management activities foreseen.

Audit: FCS_HTTPS_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) Failure to establish a HTTPS session, and reason for failure;
b) Establishment/Termination of a HTTPS session, and non-TOE endpoint of connection (IP address) for both successes and failures.

FCS_HTTPS_EXT.1 Extended: HTTPS

Hierarchical to:	No other components
Dependencies:	FCS_TLS_EXT.1 Extended: TLS
FCS_HTTPS_EXT.1.1	The TSF shall implement the HTTPS protocol that complies with RFC 2818.
FCS_HTTPS_EXT.1.2	The TSF shall implement the HTTPS protocol using TLS as specified in FCS_TLS_EXT.1.

5.1.4 FCS_TLS_EXT.1 Extended: TLS

FCS_TLS_EXT.1 Extended: TLS requires that TLS be implemented. It belongs to a new family defined for the FCS Class.

Management: FCS_TLS_EXT.1

There are no management activities foreseen.

Audit: FCS_TLS_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) Failure to establish a TLS session, and reason for failure;
b) Establishment/Termination of a TLS session, and non-TOE endpoint of connection (IP address) for both successes and failures.

FCS_TLS_EXT.1 Extended: TLS

Hierarchical to:	No other components
Dependencies:	FCS_COP.1(1) Cryptographic operation (for data encryption/decryption) FCS_COP.1(2) Cryptographic operation (for cryptographic signature)

FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)
FCS_COP.1(4) Cryptographic operation (for keyed-hash message authentication)
FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM_EXT.4 Cryptographic Key Zeroization

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols [selection: TLS 1.0 (RFC 2346), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
[selection:
None
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
].

5.1.5 FCS_RBG_EXT.1 Extended: Random Bit Generation

FCS_RBG_EXT.1 Extended: Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source. It is modeled after FCS_COP.1, but belongs to a new family defined for the FCS Class.

Management: FCS_RBG_EXT.1

There are no management activities foreseen.

Audit: FCS_RBG_EXT.1

There are no auditable events foreseen.

FCS_RBG_EXT.1 Extended: Random Bit Generation

Hierarchical to: No other components

Dependencies: None

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [selection: choose one of: NIST Special Publication 800-90 using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES), Dual_EC_DRBG (any)]; FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES] seeded by an entropy source that accumulated entropy from [selection, one or both of: a software-based noise source; a TSF-hardware-based noise source].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [selection, choose one of: 128 bits, 256 bits] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

5.1.6 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1 Password Management defines the password strength requirements that the TSF will enforce. It belongs to a new family defined for FIA class.

Management: FIA_PMG_EXT.1

There are no management activities foreseen.

Audit: FIA_PMG_EXT.1

There are no auditable events foreseen.

FIA_PMG_EXT.1 Password Management

Hierarchical to: No other components

Dependencies: None

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters: [selection: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [assignment: *other characters*]];
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;

5.1.7 FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism requires a local password-based authentication mechanism and the capability for passwords to expire. In addition, other authentication mechanisms can be specified. It is considered to be part of the FIA_UAU family.

Management: FIA_UAU_EXT.2

There are no management activities foreseen.

Audit: FIA_UAU_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) All use of the authentication mechanisms.

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

Hierarchical to: No other components

Dependencies: None

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [selection: [assignment: *other authentication mechanism(s)*], none] to perform user authentication.

5.1.8 FIA_UIA_EXT.1 Extended: Password-based Authentication and Identification Mechanism

FIA_UIA_EXT.1 Extended: Password-based Authentication and Identification Mechanism, requires a local password-based authentication mechanism and the capability for passwords to expire. In addition, other authentication mechanisms can be specified. It is based on a combination of FIA_UAU.1 and FIA_UID.1, and belongs to a new family defined for class FIA.

Management: FIA_UIA_EXT.1

There are no management activities foreseen.

Audit: FIA_UIA_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) All use of the authentication mechanism with provided user identity and origin of the attempt (e.g. IP address).

FIA_UIA_EXT.1 Extended: Password-based Authentication and Identification Mechanism

Hierarchical to:	FIA_UID.1 Timing of identification FIA_UAU.1 Timing of Authentication
Dependencies:	FTA_TAB.1
FIA_UIA_EXT.1.1	The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process: ○ Display the warning banner in accordance with FTA_TAB.1; ○ [selection: no other actions, [assignment: *list of services, actions performed by the TSF in response to non-TOE requests.*]]
FIA_UIA_EXT.1.2	The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

5.1.9 FPT_APW_EXT.1 Extended: Protection of Administrator Passwords

FPT_APW_EXT.1 Extended: Protection of Administrator Passwords requires administrator passwords to be stored in non-plaintext form and requires the TOE to prevent reading of plaintext passwords. It is modeled after FPT_SSP.2, but it belongs to a new family defined for the FPT class.

Management: FPT_APW_EXT.1

There are no management activities foreseen.

Audit: FPT_APW_EXT.1

There are no audit activities foreseen.

FPT_APW_EXT.1 Extended: Protection of Administrator Passwords

Hierarchical to:	No other components
Dependencies:	None
FPT_APW_EXT.1.1	The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2	The TSF shall prevent the reading of plaintext passwords.

5.1.10 FPT_SKP_EXT.1 Extended: Protection of TSF data (for reading of all symmetric keys)

FPT_SKP_EXT.1 Extended: Protection of TSF data (for reading of all symmetric keys) requires the TOE to prevent reading of all pre-shared, symmetric, and private keys. It is modeled after FPT_SSP.1, but it belongs to a new family defined for the FPT class.

Management: FPT_SKP_EXT.1

There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

There are no audit activities foreseen.

FPT_SKP_EXT.1 Extended: Protection of TSF data (for reading of all symmetric keys)

Hierarchical to:	No other components
Dependencies:	None
FPT_SKP_EXT.1.1	The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.1.11 FPT_TST_EXT.1 Extended: TSF testing

FPT_TST_EXT.1 Extended: TSF testing requires a suite of self-tests to be run during initial start-up in order to demonstrate correct operation of the TSF. It is modeled after FPT_TST.1, but belongs to a new family defined for class FPT.

Management: FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

There are no audit activities foreseen.

FPT_TST_EXT.1 TSF testing

Hierarchical to:	No other components
Dependencies:	None
FPT_TST_EXT.1.1	The TSF shall run a suite of self-tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

5.1.12 FPT_TUD_EXT.1 Extended: Management of TSF Data

FPT_TUD_EXT.1 Extended: Management of TSF Data, requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation. It belongs to a new family defined for the FPT class.

Management: FPT_ TUD_EXT.1

There are no management activities foreseen.

Audit: FPT_ TUD_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) Initiation of update.

FPT_ TUD_EXT.1 Extended: Trusted Update

Hierarchical to:	No other components
Dependencies:	[selection: FCS_COP.1(2) Cryptographic operation (for cryptographic signature), FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)]
FPT_TUD_EXT.1.1	The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.
FPT_TUD_EXT.1.2	The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.
FPT_TUD_EXT.1.3	The TSF shall provide a means to verify firmware/software updates to the TOE using a [selection: digital signature mechanism, published hash] prior to installing those updates.

5.1.13 FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking

FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking requires system initiated locking of an interactive session after a specified period of inactivity. It is part of the FTA_SSL family.

Management: FTA_SSL_EXT.1

The following actions could be considered for the management functions in FMT:

a) Specification of the time of user inactivity after which lock-out occurs for an individual user.

Audit: FTA_SSL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) Any attempts at unlocking an interactive session.

FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking

Hierarchical to:	No other components
Dependencies:	FIA_UIA_EXT.1 Password-based Authentication and Identification Mechanism
FTA_SSL_EXT.1.1	The TSF shall, for local interactive sessions, [selection: • lock the session – disable any activity of the user's data access display devices other than unlocking the session, and requiring that the administrator re-authenticate to the TSF prior to unlocking the session; • terminate the session] after a Security Administrator-specified time period of inactivity.

5.1.14 FFW_RUL_EXT.1 Stateful Traffic Filtering

FFW_RUL_EXT.1 Stateful Traffic Filtering requires the TOE to perform network layer 3 and 4 stateful traffic filtering. It belongs to a new class FFW.

Management: FFW_RUL_EXT.1

The following actions could be considered for the management functions in FMT:

a) Configure Firewall rules.

Audit: FFW_RUL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the ST:

a) Application of rules configured with the 'log' operation (Source and destination addresses and ports, transport layer protocol and TOE interface);

b) Indication of packets dropped due to too much network traffic, and the TOE Interface that is unable to process packets.

FFW_RUL_EXT.1 Stateful Traffic Filtering

Hierarchical to: No other components

Dependencies: None

FFW_RUL_EXT.1.1 The TSF shall perform Stateful Traffic Filtering on network packets processed by the TOE.

FFW_RUL_EXT.1.2 The TSF shall process the following network traffic protocols:

- Internet Control Message Protocol version 4 (ICMPv4)
- Internet Control Message Protocol version 6 (ICMPv6)
- Internet Protocol (IPv4)
- Internet Protocol version 6 (IPv6)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

and be capable of inspecting network packet header fields defined by the following RFCs to the extent mandated in the other elements of this SFR

- RFC 792 (ICMPv4)
- RFC 4443 (ICMPv6)
- RFC 791 (IPv4)
- RFC 2460 (IPv6)
- RFC 793 (TCP)
- RFC 768 (UDP).

FFW_RUL_EXT.1.3 The TSF shall allow the definition of Stateful Traffic Filtering rules using the following network protocol fields:

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

- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - Transport Layer Protocol
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

and distinct interface.

FFW_RUL_EXT.1.4 The TSF shall allow the following operations to be associated with Stateful Traffic Filtering rules: permit, deny, and log.

FFW_RUL_EXT.1.5 The TSF shall allow the Stateful Traffic Filtering rules to be assigned to each distinct network interface.

FFW_RUL_EXT.1.6 The TSF shall:

a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [selection: ICMP, no other protocols] based on the following network packet attributes:
   1. TCP: source and destination addresses, source and destination ports, sequence number, Flags.
   2. UDP: source and destination addresses, source and destination ports;
   3. [selection: 'ICMP: source and destination addresses, [selection: type, code, *[assignment: list of matching attributes]*]', no other protocols].

b) Remove existing traffic flows from the set of established traffic flows based on the following: [selection: session inactivity timeout, completion of the expected information flow].

FFW_RUL_EXT.1.7 The TSF shall be able to process the following network protocols:
1. FTP,
2. [selection: H.323: *[assignment: other supported protocols]*, no other protocols],

to dynamically define rules or establish sessions allowing network traffic of the following types:

- FTP: TCP data sessions in accordance with the FTP protocol as specified in RFC 959,
- [selection: *[assignment: list of additionally supported protocols and the types of network traffic to be allowed based on those protocols]*, none].

FFW_RUL_EXT.1.8 The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic:

1. *The TSF shall reject and be capable of logging packets which are invalid fragments;*

2. *The TSF shall reject and be capable of logging fragmented IP packets which cannot be re-assembled completely;*

3. *The TSF shall reject and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;*

4. *The TSF shall reject and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received;*

5. *The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a broadcast network;*

6. *The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a multicast network;*

7. *The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;*

8. *The TSF shall reject and be capable of logging network packets where the source address of the network packet is a multicast;*

9. *The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is a link-local address;*

10. *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 an address "reserved for future use" as specified in RFC 5735 for IPv4;*

11. *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" as specified in RFC 3513 for IPv6;*

12. *The TSF shall reject and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and*

13. *[selection: [assignment: other default rules enforced by the TOE], no other rules].*

FFW_RUL_EXT.1.9 When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall process the applicable Stateful Traffic Filtering rules (as determined in accordance with FFW_RUL_EXT.1.5) in the following order: administrator-defined.

FFW_RUL_EXT.1.10	When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall deny packet flow if a matching rule is not identified.

5.2 Extended TOE Security Assurance Requirement Components

There are no extended TOE Security Assurance Requirement Components.

6 SECURITY REQUIREMENTS

This section defines the Security Functional Requirements (SFRs) and Security Functional Assurance Requirements (SARs) met by the TOE. All the components have been drawn from the Network Device Protection Profile (NDPP) v1.1, and the NDPP Extended Package Stateful Traffic Filter Firewall v1.0.

6.1 Security Functional Requirements

The functional security requirements for this Security Target consist of the components from Part 2 of the CC, and those that were explicitly stated, all of which are summarized in the following table:

Table 6 – TOE Security Functional Requirements

Requirement Class	Requirement Name	Description
FAU Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User Identity Association
	FAU_STG_EXT.1	External Audit Trail Storage
FCS Cryptographic support	FCS_CKM.1	Cryptographic key generation (for asymmetric keys)
	FCS_CKM_EXT.4	Cryptographic Key Zeroization
	FCS_COP.1(1)	Cryptographic operation (for data encryption/decryption)
	FCS_COP.1(2)	Cryptographic operation (for cryptographic signature)
	FCS_COP.1(3)	Cryptographic operation (for cryptographic hashing)
	FCS_COP.1(4)	Cryptographic operation (for keyed-hash message authentication)
	FCS_HTTPS_EXT.1	Explicit: HTTPS
	FCS_RBG_EXT.1	Extended: Cryptographic Operation (Random Bit Generation)
	FCS_TLS_EXT.1	Explicit: TLS
FDP User Data Protection	FDP_RIP.2	Full Residual Information Protection
FAI Identification and Authentication	FIA_PMG_EXT.1	Password Management
	FIA_UAU.7	Protected Authentication Feedback
	FIA_UAU_EXT.2	Extended: Password-based Authentication Mechanism
	FIA_UIA_EXT.1	User Identification and Authentication
FMT Security Management	FMT_MTD.1	Management of TSF data (for general TSF data)
	FMT_SMF.1	Specification of management functions
	FMT_SMR.2	Restrictions on Security Roles
FPT Protection of the TSF	FPT_APW_EXT.1	Extended: Protection of Administrator Passwords
	FPT_SKP_EXT.1	Extended: Protection of TSF Data (for reading of all symmetric keys)
	FPT_STM.1	Reliable Time Stamps
	FPT_TST_EXT.1	TSF testing

Requirement Class	Requirement Name	Description
	FPT_TUD_EXT.1	Extended: Trusted Update
FTA TOE Access	FTA_SSL.3	TSF-initiated Termination
	FTA_SSL.4	User-initiated Termination
	FTA_SSL_EXT.1	TSF-initiated session locking
	FTA_TAB.1	Default TOE access banners
FTP Trusted Path/Channels	FTP_ITC.1	Inter-TSF Trust Channel
	FTP_TRP.1	Trusted Path
FFW Stateful Traffic Firewall	FFW_RUL_EXT.1	Stateful Traffic Filtering

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;
b) All auditable events for the [not specified] level of audit;
c) [*All administrative actions*]; and
d) [*Specifically defined auditable events listed in Table 7*]

FAU_GEN.1.2 The TSF shall record within each audit record at last the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*the information detailed in Table 7*].

Table 7 – Auditable Events

Requirements	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG_EXT.1	None	None
FCS_CKM.1	None	None
FCS_CKM_EXT.4	None	None
FCS_COP.1(1)	None	None
FCS_COP.1(2)	None	None
FCS_COP.1(3)	None	None
FCS_COP.1(4)	None	None
FCS_RBG_EXT.1	None	None
FCS_TLS_EXT.1	Failure to establish a TLS Session	Reason for failure

Requirements	Auditable Events	Additional Audit Record Contents
	Establishment/Termination of a TLS session	Non-TOE endpoint of connection (IP address) for both successes and failures
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session Establishment/Termination of a HTTPS session	Reason for failure Non-TOE endpoint of connection (IP address) for both successes and failures
FDP_RIP.2	None	None
FIA_PMG_EXT.1	None	None
FIA_UIA_EXT.1	All use of the identification and authentication mechanism	Provided user identity, origin of the attempt (e.g., IP address)
FIA_UAU_EXT.2	All use of the authentication mechanism	Origin of the attempt (e.g., IP address)
FIA_UAU.7	None	None
FMT_MTD.1	None	None
FMT_SMF.1	None	None
FMT_SMR.2	None	None
FPT_SKP_EXT.1	None	None
FPT_APW_EXT.1	None	None
FPT_STM.1	Changes to the time	The old and new values for the time Origin of the attempt (e.g. IP address)
FPT_TUD_EXT.1	Initiation of update	No additional information
FPT_TST_EXT.1	None	None
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session	No additional information
FTA_SSL.3	The termination of a remote session by the session locking mechanism	No additional information
FTA_SSL.4	The termination of an interactive session	No additional information
FTA_TAB.1	None	None
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions	Identification of the claimed user identity
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions	Identification of the initiator and target of failed trusted channels establishment attempt
FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface
	Indication of packets dropped due to too much network traffic	TOE Interface that is unable to process packets

6.1.1.2 FAU_GEN.2 User Identity Association

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to

associate each auditable event with the identity of the user that caused the event.

6.1.1.3 FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.1.1 The TSF shall be able to [transmit the generated audit data to an external IT entity] using a trusted channel implementing the [TLS] protocol.

6.1.2 Cryptographic Support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1 **Refinement:** The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance with
[
- *NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes*]

and specified cryptographic key sizes [*equivalent to, or greater than, a symmetric key strength of 112 bits*].

6.1.2.2 FCS_CKM_EXT.4 Cryptographic Key Zeroization

FCS_CKM_EXT.4.1 The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

6.1.2.3 FCS_COP.1(1) Cryptographic Operation (for data encryption/decryption)

FCS_COP.1.1(1) The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [*AES operating in [**CBC**[6]]*] and cryptographic key sizes [*128-bits and 256-bits*] that meets the following: [
- ***FIPS PUB 197, "Advanced Encryption Standard (AES)"***
- ***[NIST SP 800-38A,]***]

6.1.2.4 FCS_COP.1(2) Cryptographic operation (for cryptographic signature)

FCS_COP.1.1(2) The TSF shall perform [*cryptographic signature services*] in accordance with a **[RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater]**
that meets the following:
- **FIPS PUB 186-2 or FIPS PUB 186-3, "Digital Signature Standard"**

6.1.2.5 FCS_COP.1(3) Cryptographic Operation (for cryptographic hashing)

FCS_COP.1.1(3) The TSF shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm **[SHA-1, SHA-256] and message digest sizes**

[6] CBC: Cipher Block Chaining

[160, 256] bits that meet the following: [*FIPS Pub 180-3, "Secure Hash Standard."*]

6.1.2.6 FCS_COP.1(4) Cryptographic Operation (for keyed-hash message authentication)

FCS_COP.1.1(4) The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm HMAC-**[SHA-1, SHA-256], key size [160-bit, 256-bit], and message digest sizes [160, 256] bits** that meet the following: [*FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."*]

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

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [NIST Special Publication 800-90 using [CTR_DRBG (AES)] seeded by an entropy source that accumulated entropy from [a TSF-hardware-based noise source].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [256 bits] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

6.1.2.8 FCS_HTTPS_EXT.1 Extended: HTTPS

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement the HTTPS protocol using TLS as specified in FCS_TLS_EXT.1.

6.1.2.9 FCS_TLS_EXT.1 Extended: TLS

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols [TLS 1.0 (RFC 2346), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
[TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256].

6.1.3 User Data Protection (FDP)

6.1.3.1 FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [allocation of the resource to] all objects.

6.1.4 Identification and Authentication (FIA)

6.1.4.1 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”];
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;

6.1.4.2 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [Query the TOE version, Zeroize the TOE via local console]

FIA_UIA_EXT.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

6.1.4.3 FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [none] to perform user authentication.

6.1.4.4 FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1 The TSF shall provide only [*obscured feedback*] to the administrative user while the authentication is in progress at the local console.

6.1.5 Security Management (FMT)

6.1.5.1 FMT_MTD.1 Management of TSF Data (for general TSF data)

FMT_MTD.1.1 The TSF shall restrict the ability to [*manage*] the [*TSF data*] to [*the Security Administrators*].

6.1.5.2 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [

- *Ability to administer the TOE locally and remotely;*
- *Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates*[7]*;*
- *[Ability to configure the cryptographic functionality;*
- *Ability to configure firewall rules*]

[7] FortiOS requires that the candidate firmware upgrade be uploaded to the TOE prior to the digital signature being validated

6.1.5.3 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles: [*Authorized Administrator*]

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions [

- *Authorized Administrator role shall be able to administer the TOE locally;*
- *Authorized Administrator role shall be able to administer the TOE remotely*;]

are satisfied.

6.1.6 Protection of the TSF (FPT)

6.1.6.1 FPT_SKP_EXT.1 Extended: Protection of TSF Data (for reading of all symmetric keys)

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

6.1.6.2 FPT_APW_EXT.1 Extended: Protection of Administrator Passwords

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

6.1.6.3 FPT_STM.1 Reliable Time Stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

6.1.6.4 FPT_TUD_EXT.1 Extended: Trusted Update

FPT_TUD_EXT.1.1 The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3 The TSF shall provide a means to verify firmware/software updates to the TOE using a [digital signature mechanism] prior to installing those updates.

6.1.6.5 FPT_TST_EXT.1: TSF Testing

FPT_TST_EXT.1.1 The TSF shall run a suite of self-tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

6.1.7 TOE Access (FTA)

6.1.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [terminate the session] after a Security Administrator-specified time period of inactivity.

6.1.7.2 TSF-initiated Termination

FTA_SSL.3.1 **Refinement**: The TSF shall terminate **a remote** interactive session after a [*Security Administrator-configurable time interval of session inactivity*].

6.1.7.3 FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1 The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.

6.1.7.4 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1 **Refinement**: Before establishing **an administrative user** session the TSF shall display **a Security Administrator-specified advisory notice and consent** warning message regarding use of the TOE.

6.1.8 Trusted Path/Channels (FTP)

6.1.8.1 FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 **Refinement**: The TSF shall **use [TLS]** to provide a **trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: [audit server, authentication server]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data **from disclosure and detection of modification of the channel data**.

FTP_ITC.1.2 The TSF shall permit [the TSF] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [logging of audit messages, verification of user credentials].

6.1.8.2 FTP_TRP.1 Trusted Path

FTP_TRP.1.1 **Refinement**: The TSF shall **use [TLS/HTTPS]** provide a **trusted** communication path between itself and **remote administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [disclosure and **detection of modification of the communicated data**].

FTP_TRP.1.2 **Refinement**: The TSF shall permit **remote administrators** to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [initial administrator authentication and *all remote administration actions*].

6.1.9 Stateful Traffic Filtering (FIA)

6.1.9.1 FFW_RUL_EXT.1 Stateful Traffic Filtering

FFW_RUL_EXT.1.1 The TSF shall perform Stateful Traffic Filtering on network packets processed by the TOE.

FFW_RUL_EXT.1.2 The TSF shall process the following network traffic protocols:

- Internet Control Message Protocol version 4 (ICMPv4)
- Internet Control Message Protocol version 6 (ICMPv6)
- Internet Protocol (IPv4)
- Internet Protocol version 6 (IPv6)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

and be capable of inspecting network packet header fields defined by the following RFCs to the extent mandated in the other elements of this SFR

- RFC 792 (ICMPv4)
- RFC 4443 (ICMPv6)
- RFC 791 (IPv4)
- RFC 2460 (IPv6)
- RFC 793 (TCP)
- RFC 768 (UDP).

FFW_RUL_EXT.1.3 The TSF shall allow the definition of Stateful Traffic Filtering rules using the following network protocol fields:

- ICMPv4
  - Type
  - Code
- ICMPv6
  - Type
  - Code
- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - Transport Layer Protocol
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

and distinct interface

FFW_RUL_EXT.1.4 The TSF shall allow the following operations to be associated with Stateful Traffic Filtering rules: permit, deny, and log.

FFW_RUL_EXT.1.5 The TSF shall allow the Stateful Traffic Filtering rules to be assigned to each distinct network interface.

FFW_RUL_EXT.1.6 The TSF shall:

a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [ICMP] based on the following network packet attributes:
   1. TCP: source and destination addresses, source and destination ports, sequence number, Flags.
   2. UDP: source and destination addresses, source and destination ports;
   3. [ICMP source and destination addresses, [type, code]].
b) Remove existing traffic flows from the set of established traffic flows based on the following: [session inactivity timeout, completion of the expected information flow].

FFW_RUL_EXT.1.7 The TSF shall be able to process the following network protocols:
FTP,[H.323, no other protocols], to dynamically define rules or establish sessions allowing network traffic of the following types:

- FTP: TCP data sessions in accordance with the FTP protocol as specified in RFC 959,
- [H.323: Inspection of UDP SIP traffic protocol as specified in RFC 4123].

FFW_RUL_EXT.1.8 The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic:

1. *The TSF shall reject and be capable of logging packets which are invalid fragments;*

2. *The TSF shall reject and be capable of logging fragmented IP packets which cannot be re-assembled completely;*

3. *The TSF shall reject and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;*

4. *The TSF shall reject and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received;*

5. *The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a broadcast network;*

6. *The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a multicast network;*

7. *The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;*

8. *The TSF shall reject and be capable of logging network packets where the source address of the network packet is a multicast;*

9. *The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is a link-local address;*

10. *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 an address "reserved for future use" as specified in RFC 5735 for IPv4;*

11. *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" as specified in RFC 3513 for IPv6;*

12. *The TSF shall reject and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and*

13. *[no other rules].*

FFW_RUL_EXT.1.9 When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall process the applicable Stateful Traffic Filtering rules (as determined in accordance with FFW_RUL_EXT.1.5) in the following order: administrator-defined.

FFW_RUL_EXT.1.10 When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall deny packet flow if a matching rule is not identified.

6.2 Security Assurance Requirements

This section defines the Security Assurance Requirements (SARs) for the TOE. The assurance requirements are taken from NDPP v1.1 and are the EAL 1 components as specified in Part 3 of the CC. The assurance components are summarized in the following table:

Table 8 – Security Assurance Requirements

Assurance Classes	Assurance Component	Description
Security Target	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives

	ASE_REQ.1	Security requirements
	ASE_TSS.1	TOE summary specification
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative Procedures
Lifecycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests	ATE_IND.1	Independent Testing – Conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

7 TOE SUMMARY SPECIFICATION

This section presents information to detail how the TOE meets the security functional requirements described in previous sections of this ST.

7.1 Security Audit

For all administrative actions including management of the TOE authentication is required before any actions can occur on the TOE. When an action identified in table 7 is triggered the TOE will write the event including the administrative username of the user triggering the event to the audit log.

In the evaluated configuration the event log is always considered to be on and logging to the remote audit server once the TOE is fully initialized and services are available in normal operation. The TOE logs the startup and shutdown of the TOE, and this can be considered to be equivalent to the startup and shutdown of the audit system.

The TOE is capable of logging the audit messages both locally and remotely, and has configurable actions when the audit logs are filled. By default the TOE will log locally and will block further traffic from occurring should the local storage become exhausted. Guidance is provided to the administrator to modify this behavior to overwrite the oldest audit logs upon hitting a threshold of memory capacity. The most recent audit records are stored locally; using memory, a hard disk or a FLASH memory card depending on the model.

The TOE also has configurable options for the remote storage of the audit events. These events are sent in real-time to one or more capable audit servers. In the evaluated configuration these audit servers consist of the FortiAnalyzer analytics suite secured through TLS. Details as to the placement of this component of the IT environment is provided in Figure 1 of this document. Audit events are sent in real-time meaning that there is no capacity for the TOE store or buffer any system events that might occur while the audit server is offline.

The TOE is capable of logging messages to the audit log for interactions which occur on the remote interfaces. These events include attempts to establish or terminate sessions and errors detected during decryption or validation of data.

The list of firewall rules is consulted in the order listed by the TOE on the CLI or GUI. Once a set of matching security attributes is found in the firewall rules the action identified by this rule is taken. Should this action include logging of the packet, the following information is written to the audit server:

- Transportation Layer Protocol
- Source IP address
- Source Port
- TOE Interface
- Destination IP address
- Destination Port

A number of factors can impact the performance of the Fortinet appliances and their ability to mediate traffic between networks. FortiGate™ appliances are rated for their capacity to forward and filter traffic during normal operations. Should this capacity be exceeded the packet processing engine will write an audit event to the audit log. The TOE will then drop packets on the affected interface until such a time that there are sufficient resources to resume inspection of packets and information flows.

7.2 Cryptographic Support

The TOE uses FIPS-approved cryptography that has been implemented in FIPS 140-2 validated cryptographic modules. The FIPS-validated cryptographic modules implemented in the TSF meet Security Level 1 overall and meet Security Level 2 for the following: cryptographic module ports and interfaces; roles, services and authentication; and design assurance. The proprietary FortiASIC™ chip is a hardware component which forms part of the validated cryptographic modules used in the TOE. Cryptographic key destruction by the TOE meets the key zeroization requirements of Key Management Security Level 1 from FIPS PUB 140-2. The TOE only stores keys in memory, either in RAM or Flash memory. Keys are destroyed by overwriting the key storage area with an alternating pattern at least once.

The TOE is capable of generating 256 bits of entropy using a dedicated hardware noise source and using this to seed the random bit generator in order to provide cryptographic services with up to 256 bits of strength. The TOE is also capable of importing cryptographic keys and certificates from outside the TOE boundary. These keys are zeroized when no longer required and the TOE offers a function to zeroize these keys on demand.

A detailed design of the cryptographic subsystems and entropy noise sources provided by the TOE has been conducted and was used to design the TOE to ensure strong seeding of the DRBG. This has been found to meet the entropy requirements for collection, strength and the seeding of the DRBG contained within the TOE.

The following certificates have been issued by the CMVP and CAVP for FortiOS 5.0 and are implemented accordingly in the TOE.

Table 9 – Software Cryptographic Module Algorithms

Module	Cryptographic Operations	Cryptographic Algorithim	Key Sizes (bits)	Message Digest Size	Standards Compliance	Certificate #
FortiOS FIPS[9]	Symmetric Encryption and Decryption	Triple-DES operating in CBC	128, 192	N/A	FIPS 46-3	CAVP Certificate # 1807
		AES operating in CBC	128, 256	N/A	FIPS PUB 197 (AES) NIST SP800-38A	CAVP Certificate # 3169
	Cryptographic Hashing	SHA-1, SHA-256	160, 256	160, 256	FIPS Pub 198-1 (HMAC) FIPS Pub 180-3 (SHS)	CAVP Certificate # 2622
	Keyed-Hash message authentication (HMAC)	SHA-1, SHA-256	160, 256	160, 256	FIPS Pub 180-3 (SHS)	CAVP Certificate # 1997
	Deterministic Random Bit	AES 256 CTR-DRBG			SP 800-90	CAVP Certificate #652

[9] The “FIPS” module from Fortinet consists of the FortiASIC and is used for acceleration of encryption where possible and applicable. Both the “FIPS” and “SSL” modules have been CAVP tested.

	Generator (DRBG)					
FortiOS SSL [10]	Symmetric Encryption and Decryption	Triple-DES operating in CBC	128, 192	N/A	FIPS 46-3	CAVP Certificate # 1808
		AES operating in CBC	128, 256	N/A	FIPS PUB 197 (AES) NIST SP800-38A	CAVP Certificate # 3171
	Cryptographic Hashing	SHA-1, SHA-256	160, 256	160, 256	FIPS Pub 198-1 (HMAC) FIPS Pub 180-3 (SHS)	CAVP Certificate # 2624
	Keyed-Hash message authentication (HMAC)	SHA-1, SHA-256	160, 256	160, 256	FIPS Pub 180-3 (SHS)	CAVP Certificate # 1999
	Signature Verification	rDSA	1024, 2048, 3072	N/A	FIPS PUB 186-4 (DSS)	CAVP Certificate # 1607
	Signature Generation	rDSA	2048, 3072	N/A	FIPS PUB 186-4 (DSS)	CAVP Certificate # 1607

[10] While the name of this module is “SSL” it is used for all operations which are not available on the ASIC as well as well as supporting the Web GUI. This module has undergone CAVP validation and corresponds with the name provided on CAVP certificate.

Table 10 – FortiOS Hardware Cryptographic Certificates

Module	Cryptographic Operations	Cryptographic Algorithm	Key Sizes (bits)	Message Digest Size	FIPS Standard	Certificate #
FortiASIC CP6	Symmetric Encryption and Decryption	Triple-DES operating in CBC	128, 192	N/A	FIPS 46-3	CAVP Certificate # 1804
		AES operating in CBC	128, 256	N/A	FIPS PUB 197 (AES) NIST SP800-38A	CAVP Certificate # 3166
	Cryptographic Hashing	SHA-1, SHA-256	160, 256	160, 256	FIPS Pub 198-1 (HMAC) FIPS Pub 180-3 (SHS)	CAVP Certificate # 2619
	Keyed-Hash message authentication (HMAC)	SHA-1, SHA-256	160, 256	160, 256	FIPS Pub 180-3 (SHS)	CAVP Certificate # 1994
	Signature Verification	rDSA	1024, 2048	N/A	FIPS PUB 186-4 (DSS)	CAVP Certificate # 1604
	Signature Generation	rDSA	2048	N/A	FIPS PUB 186-4 (DSS)	CAVP Certificate # 1604
FortiASIC CP7	Symmetric Encryption and Decryption	Triple-DES operating in CBC	128, 192	N/A	FIPS 46-3	CAVP Certificate # 1805
		AES operating in CBC	128, 256	N/A	FIPS PUB 197 (AES) NIST SP800-38A	CAVP Certificate # 3167
	Cryptographic Hashing	SHA-1, SHA-256	160, 256	160, 256	FIPS Pub 198-1 (HMAC) FIPS Pub 180-3 (SHS)	CAVP Certificate # 2620
	Keyed-Hash message authentication (HMAC)	SHA-1, SHA-256	160, 256	160, 256	FIPS Pub 180-3 (SHS)	CAVP Certificate # 1995
	Signature Verification	rDSA	1024, 2048	N/A	FIPS PUB 186-4 (DSS)	CAVP Certificate # 1605
	Signature Generation	rDSA	2048	N/A	FIPS PUB 186-4 (DSS)	CAVP Certificate # 1605

Module	Cryptographic Operations	Cryptographic Algorithm	Key Sizes (bits)	Message Digest Size	FIPS Standard	Certificate #
FortiASIC CP8	Symmetric Encryption and Decryption	Triple-DES operating in CBC	128, 192	N/A	FIPS 46-3	CAVP Certificate # 1806
		AES operating in CBC	128, 256	N/A	FIPS PUB 197 (AES) NIST SP800-38A	CAVP Certificate # 3168
	Cryptographic Hashing	SHA-1, SHA-256	160, 256	160, 256	FIPS Pub 198-1 (HMAC) FIPS Pub 180-3 (SHS)	CAVP Certificate # 2621
	Keyed-Hash message authentication (HMAC)	SHA-1, SHA-256	160, 256	160, 256	FIPS Pub 180-3 (SHS)	CAVP Certificate # 1996
	Signature Verification	rDSA	1024, 2048	N/A	FIPS PUB 186-4 (DSS)	CAVP Certificate # 1606
	Signature Generation	rDSA	2048	N/A	FIPS PUB 186-4 (DSS)	CAVP Certificate # 1606

As part of the CMVP testing the FIPS lab validated the Diffie-Hellman Ephemeral (DHE) implementation for both the “FIPS” and “SSL” cryptographic modules via the CAVS tool for the correctness of the implementation as both an initiator and responder. For additional details on Diffie-Hellman or any other cryptographic operations, the author is encouraged to reference the appropriate certificate as stated in Table 9 and Table 10.

For cryptographic operations the TOE will leverage the ASIC for cryptographic operations where possible based on the ciphers and algorithms present on each of the ASIC chips. In the evaluated configuration the only operations accelerated in this manner are the TLS connections. For additional details on the ASIC in each TOE model as well as the ciphers offered by that ASIC the FIPS validation listed in Table 9.

Should the cryptographic algorithm not be present on the ASIC or the ASIC become unavailable the TOE will fall back to the slower SSL module listed in Table 10. Both cryptographic modules have had their methods CAVP validated and the secure operation of the TOE will be maintained.

7.2.1 Entropy Source and Random Bit Generation

The TOE implements an entropy collection system from a hardware based FortiTRNG™ noise source which is derived from wide-band RF white noise implemented through a USB interface. The raw unconditioned noise from this source is pooled and conditioned prior to being used by a FIPS approved

DRBG. The FortiOS kernel has been customized to take advantage of this strong random bit generator for all calls to the kernel as well as for the cryptographic operations listed in FCS_COP.1.

The noise source has been analyzed by a 3rd party using a Fortinet supplied tool to collect raw noise prior to any conditioning. This data was analyzed using a NIAP supplied entropy test tool which confirmed that the entropy present in the aggregate samples was found to demonstrate this claim during typical operating conditions. This noise source was found to be capable of providing entropy to the random number generator to produce random values with up to 256 bits of security strength.

The Fortinet Cryptographic Module contains a CTR_DRBG implemented per NIST SP 800-90A and is seeded with an entropy source described above. Entropy from the noise source is extracted and used to seed the DRBG with at least 256 bits of entropy. As a failure of the entropy source is a blocking event for the cryptographic system and is continually monitored for health this helps ensure that only a catastrophic failure of the noise source will halt the operation of the TOE. This helps ensure smooth ongoing operation under typical conditions. The CTR_DRBG implementation has been CAVP tested to ensure correct operation.

7.2.2 Cryptographically Trusted Paths

Trusted paths are used to protect remote administrator authentication and all remote administrator actions. Remote administration sessions apply to the Network Web-Based GUI and Network CLI.

7.2.3 HTTPS

The Network Web-Based GUI uses the HTTPS protocol for secure administrator communications. With respect to the TOE implementation of HTTPS, TLS version 1.0 (RFC 2246), 1.1 (RFC 2246) and 1.2 (RFC 4346) can be used to encrypt and authenticate administration sessions between the remote browser and TOE. The TOE supports ciphersuites; TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA, TLS_DHE_RSA_WITH_AES_128_CBC_SHA, TLS_DHE_RSA_WITH_AES_256_CBC_SHA, TLS_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_256_CBC_ SHA256, TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256, TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256. TLS 1.0 is also used for the purposes of protecting the audit logs while in transit to the FortiAnalyzer or syslog audit servers.

7.2.4 TLS

The TLS ciphersuites mean that the keying material is determined when the session is established through a Diffie-Hellman (DH) exchange which consists of:

- Server sends 2048-bit RSA public certificate
- Server generates, signs (RSA PKCS#1) and sends DH parameters and DH public value
- Client generates and sends DH public value. The keying material is then used to encrypt/decrypt (AES128 and AES256) and authenticate (HMAC-SHA1 or HMAC-SHA256) the data exchange.

When a connection is first established, the server presents the 2048-bit RSA certificate to the connecting web browser. The administrator can examine the certificate to validate the identity of the TOE and then choose to continue with the connection if the certificate conforms to the expected values. Only after the certificate has been explicitly accepted as valid does the above TLS 1.0, 1.1 or 1.2 authentication with the administrator's web browser occur with the TOE to establish the trusted channel. After this channel

is established the administrator will be presented with the login page using this channel and HTTP, where the user and password credentials can be submitted for administrator authentication.

When the TOE uses TLS 1.0 for the purposes of protecting the audit logs during transit over the network the TOE will negotiate an appropriate cipher suite based on the approved list of ciphers. Audit logs are sent in real time and for each auditable event which is to be written to the audit server the TOE will verify that the RSA certificate present on the audit server matches the certificate which was presented when the TOE was registered with the audit server. The TOE then negotiates a suitable claimed cipher with the audit server, generates and sends the DH public value for keying and encrypts the audit message which is to be sent to the TOE.

The trusted channels provide communication between the TOE and the FortiSSO and FortiAnalyzer units. These channels are logically distinct from other communication channels and provide assured identification of the end points and protection of the channel data from disclosure. A FortiAnalyzer unit may be used to collect and analyze logging information.

7.2.5 Cryptographic Self Tests and TOE Update Integrity

The TSF provides a cryptographic function that an administrator may use to verify the integrity of the TSF executable code. During a normal boot-up sequence the TOE administrator can see on the local console the following types of tests in order:

- Configuration file tests
- FIPS AES, SHA, 3DES and RSA tests
- Firmware integrity tests
- Entropy tests
- RNG tests

Indication of successful tests would appear as follows:

Running <test>... passed

Completion of all self-tests is indicated by:

Self-tests passed

The TOE is capable of running these tests at the request of an administrator, and periodically at an administrator-specified interval not less than once a day to demonstrate the correct operation of the cryptographic components of the TSF. The TOE will enter into a FIPS Error Mode when failure of a self-test (integrity verification self-test, or cryptographic self-test) is detected. This mode allows the TOE to enter into a secure state. These self-tests are executed on initial start-up or at the request of an administrator. Upon successful completion of these tests an audit log will be generated by the TOE and sent to the remote audit server.

The TOE provides a USB interface which may be used by an authorized administrator to load private keys from a USB token. For example the 2048-bit RSA certificate used by the Network Web-Based GUI can be replaced by certificates trusted by an authorized administrator. These keys/certificates are to be placed on the USB token and the load operation can be executed via a Network CLI or Network Web-Based GUI administrator session.

7.2.6 Conformance to NIST SP800-56

The TOE fulfills all of the NIST SP 800-56B requirements without extensions. The TOE does not implement any functionality within this standard that is listed as “should not” and “shall not”.

Specifically the TOE claims conformance to 5.1 (Cryptographic Hash Functions), 5.2 (Message Authentication Code Algorithm), 5.3 (Random Bit Generation), 5.4 (Prime Number Generators), 5.5 (Primality Testing Methods), 5.6 (Nonces), 5.9 (Key Derivation Functions for Key Establishment Schemes), 6.1 (RSA Key Pairs - General Requirements), 6.2 (Criteria for RSA Key Pairs for Key Establishment), 6.3 (RSA Key Pair Generators), 6.4 (Assurance of Validity), 6.5 (Assurance of Private Key Possession), 6.6 (Key Confirmation), and 8 (Key Agreement Schemes). The TOE complies with RSA key pair generation according to FIPS 186-2 and FIPS 186-3 in SP 800-56B.

7.2.7 Key and CSP storage and zeroization

The TOE maintains a number of keys and CSPs related to its secure operation. Administrative passwords are stored in the configuration file on the flash drive of the TOE and are encoded via a hash function to ensure their confidentiality. These keys are capable of being zeroized either through a format of the flash memory (as described in the Fortinet 5.0 Level 1 Security Policy) or through a factory reset of the TOE.

Certificates for the purposes of HTTPS andTLS connections are maintained on the flash filesystem and are not viewable through the TOE interfaces. When these keys are no longer required the administrator can remove the keys through the formatting of the flash memory. Details on this process are contained in the FIPS level 1 security policy of the TOE.

Additionally the TOE stores a number of CSPs in volatile memory during normal operation of the FortiOS SSL and FortiOS FISP cryptographic modules. These CSPs include the ephemeral keys and copies of the persistent keys described above are loaded into memory during normal operation. The TOE maintains these keys in its volatile memory in order to support theTLS and HTTPS connections to the TOE. These CSPs include:

- The RSA signature generation key
- The RSA private key decryption key
- AES encryption/decryption key
- AES CMAC generation/verification key
- HMAC Key
- Diffie-Hellman Private agreement key
- RNG Seed

These CSPs are cleared when the process terminates. Each of the CSPs are protected from unauthorized access via the FortiOS memory management which disallows any memory reads from other processes within the OS ensuring that the CSPs are only available to the calling application.

7.3 User Data Protection

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 done through the zeroization of data when the memory structure is initially created and strict bounds checking on the data prior to it being assigned in memory.

7.4 Identification and Authentication

The TOE supports a variety of methods of Identification and Authentication to both local and external sources. Regardless of the method of administration that is chosen by the administrator no administrative action is possible prior to authentication.

The TOE uses a local password database for all of its credentials by default. Authorized administrators are able to configure which of these features and functionality the TOE will use when authenticating against a TSFI. Passwords can be created through the usage of mixed case characters, digits and the special characters "!", "@", "#", "$", "%", "^", "&", "*", "(" and ")".By default the TOE requires at least 6 printable charicters and 8 characters minimum for passwords. This value can be overwritten by the administrator to suit the needs of their environment.

When a remote authentication source is enabled and a request for access is made from on one of the TSFI's the TOE will consult the configured authentication providers. The credentials are then checked against the credential database through a cryptographically protected channel. If the remote credential store returns a successful authentication the administrator is authenticated to the TOE. Once authenticated the administrator is granted access to the TSFI and is subject to the standard enforcement practices for management of the TOE data and management of the TSF.

7.4.1 Web/HTTPS

By default the web/HTTPS interface is enabled on the TOE WAN port. The TOE may also be configured to allow or disallow access to this TSFI on a per-network port basis in either the CLI or the web UI. The HTTPS web interface is accessed by going to the TOE IP on port 443. Once connected to the port and the HTTPS session is established the TOE provides a warning banner according to FTA_TAB.1 which the administrator must accept prior to proceeding. Following this banner the user will then accept the warning and be presented with a username and login screen. The administrator will then provide their credentials which are accepted by the webserver and protected in transit. Once the authentication has been received the local credential store is consulted for a match. Should there be a match in the local password database access is granted to the TOE. During the authentication process the user's password is entered in a "password" input box. A failed authentication attempt will be met with the '' Authentication failure. Please try again..." error message.

Successful authentication may be implemented in two ways. The administrator may be met with a post authentication warning forcing them to accept another warning. If this configuration option is not enabled the main login dashboard will be presented. By default this dashboard will contain the hostname, serial number and a number of other pieces of information regarding the TOE.

7.4.2 Local Serial Console

The local console is only accessible through the use of the dedicated management port present on the TOE and requires that the management station be appropriately configured. Depending on the hardware model this could be via Serial Console or USB. The reader is encouraged to consult the guidance documentation for details as to which method is supported by their specific model. For additional details regarding the software requirements for the IT environment please see section 1.5.3

By default the local access is enabled and may not be disabled Authentication over this interface is the same as over the secure shell. First the pre-login warning banner is displayed as configured by the administrator. Next the user is prompted for their username which is echoed back to the screen. Following the identification the user is requested to put in their password which is hidden and provides no feedback indicating any progress. Once the authentication has been received the local credential store is consulted and if there is a match access is granted to the TOE. Following successful authentication a post-login warning may be seen and the TOE will change the command prompt to the hostname followed by #. During the boot process the TOE can be zeroized through the usage of the boot menu and selecting "format boot device". This process does not require authentication.

7.4.3 Universal Serial Bus

Some models provide a local console over USB and a native client application present in the TOE environment. Once the FortiExplorer™ application is installed the Administrator must connect the TOE to the management station using a supplied USB cable. This interface will then query the TOE version to ensure that it is compatible with the TOE and then will prompt for authentication. This interface allows for basic configuration, such as admin password, WAN and LAN settings, and device registration. Further guidance on configuring the TOE to put it into the evaluated configuration is provided in the guidance supplement. Once the basic network configuration has been completed future requests to the TOE for information or network services will be done over the GUI interface.

7.5 Security Management

The security management for the TOE is implemented on a per-interface basis. Regardless of the interface no management functionality is possible prior to authentication. The TOE is capable of having custom roles defined however, only the *Authorized Administrator* role who can administer all functionality of the TOE is defined.

7.5.1 Local Console CLI

The CLI requires identification and authentication prior to any administrative session being established with the TOE. Sessions are terminated after inactivity to ensure that stale sessions may not be hijacked through physical access to the serial port or through an unattended administrator workstation. Any attempt by an administrator to access the CLI without a valid session will be rejected and the administrator will be forced to authenticate.

Once authenticated the CLI gives administrators full control over all aspects of the TOE including the management and setting of users, firewall rules and cryptographic operations. Additionally when a Security Administrator logs into the TOE via the web interface they are able to manage the firewall rules for the TOE.

7.5.2 Web UI

The TOE tracks administrative sessions on the WebUI through the use of cookies and a session database on the TOE. When an administrator logs onto the TOE the cookies and session database are consulted to determine if there is already an open session for this instance. In the event that there is no pre-existing session established for the management of the TOE the user is redirected to the login page. Stale administrative sessions are logged out after a period of inactivity to ensure that unattended administrator sessions can't be hijacked.

Once authenticated the CLI gives administrators full control over all aspects of the TOE including the management and setting of users, firewall rules and cryptographic operations. Additionally it can allow Security Administrators the ability to manage the firewall rules of the TOE.

7.6 Protection of the TSF

The FortiGate™ appliances use a number of methods to protect themselves and the communications channels which it provides from potentially hostile entities. This includes an internal clock source provided by the kernel of the TOE which allows the auditable events to be reviewed in a reliable manner to reproduce the sequence of events that was observed.

7.6.1 Cryptographic Key and Password Storage

The TOE itself is a FIPS 140-2 level 1 cryptographically validated module. This means that it has a number of physical security protections in place including but not limited to the protection of any keys provided to or stored within the cryptographic module. Cryptographic keys within this module are generated and destroyed per the FIPS guidelines and are not capable of being viewed through the CLI or Web interface. The TOE does not provide any method of direct access to view or modify files over either of these interfaces.

Cryptographic certificates related to the HTTPS interface are stored on the local filesystem of the TOE. An authorized administrator can generate a certificate signing request from the TOE and import the signed certificate back into the TOE. Once this CSP is imported into the TOE this information cannot be viewed again through any of the TSFI's. These keys can be zeroized through the methods described in section 7.2.9.

Pre-shared keys related to administrator passwords and other credentials for the secure operation of the TOE are stored in the TOE's configuration file. Authorized administrators are allowed to enter this information through the communications paths such as the local console or HTTPS GUI. Once the password is entered the TOE encrypts the password using AES-128 and writes the password to the configuration file permanently obscuring the contents. This 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. The AES key for the protection of this configuration file and its passwords is generated by the TOE when the TOE is initialized and put into FIPS mode.

7.6.2 FortiGate™ Product Updates

The TOE protects itself during updates through the use of a cryptographic signature. The update process goes as follows. The administrator downloads the TOE to their workstation from https://support.fortinet.com. The administrator can then verify the integrity of the update by initiating the update process. To do this the administrator will then copy the file to the TOE via a trusted path such as the HTTPS web interface.

Once the firmware update is uploaded to the TOE a 2048 bit RSA signature is verified for any TOE firmware build. The signature is compared to a known key value stored on the TOE and hardcoded into the firmware image. Before proceeding with a firmware upgrade via the GUI or CLI, the following process is followed when in the evaluated mode of operation:

- If signature is not present-> abort upgrade
- Extract public key and signature from the firmware

- Validate that public key is same as is stored previously on the TOE. If the public keys do not match abort the upgrade.
- Validate image signature using public key from the update. If the image validation using the public key fails abort upgrade.

If the firmware load test fails, the error message displayed is “File is not an update file.” Otherwise the TOE displays “upgrade successful” and reboots.

7.6.3 Self-Tests

The TOE performs a number of self-tests at start-up and on an ongoing basis. At startup the TOE undergoes the following tests in order:

- 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 TOE is also capable of performing the following tests on-demand

- FIPS 140-2 KAT (as described above)
- Noise source (as described above)

The TOE also performs the following ongoing self-tests

- Noise source pattern analysis

7.7 TOE Access

The TOE has a number of methods to restrict access to only those administrators who are authorized to administer the TOE. The first is a login warning prior to allowing a user to log in stating that this is a restricted access system and only authorized administrators should attempt to login. This prompt is present on the local console as well as the HTTPS interface.

The TOE also provides a method for both local and remote sessions to be protected in the event of an Administrator leaving their session unattended. An authorized administrator can configure the TOE to terminate inactive local and remote sessions following a specified period of time. By default in the evaluated configuration this timeout value is the same and it is set to 5 minutes. Finally should an administrator wish to terminate their session the TOE is able to terminate their session from the TOE side. On the local console and the HTTPS GUI the user session is terminated and the user is taken back to the warning banner stating that this is a restricted access system which they are forced to accept prior to going to the login page.

7.8 Trusted Path/Channels

The TOE is designed for secure operation from a trusted administrator to ensure correct operation of the stateful traffic firewall implemented by the TOE. Additionally the TOE can secure communications to a remote FortiAnalyzer audit server through TLS.

A trusted path for an administrator to communicate with the TOE is implemented through the use of a HTTPS GUI. When in the evaluated configuration this is the only method of remote communication which is possible with the TOE. When a remote administrator initiates a connection on this interface the TOE will respond with a cryptographically strong communication path using TLS 1.0, 1.1 or 1.2 to the workstation of the administrator which will be used for all communication between the TOE and the authorized administrator. The TOE will detect, log and reject any packets which indicate that the communications of this path have been tampered with or modified.

Trusted channels are provided for the purposes of securing remote authentication and the storage of audit logs. When the TOE is configured to send the logs to FortiAnalyzer communications are secured by TLS 1.0. When an auditable event which is required to be written to the remote audit server is generated the TOE connects to the audit server over TLS and writes the event log to the audit server. No persistent connection is maintained with the audit server.

In the event of a successful authentication the administrator will be assigned their appropriate permissions on the TSFI and be granted access for that session. In the event that the user fails their authentication the TOE will log an audit log stating that there was a failure to authenticate.

The TOE is capable of detecting modification or tampering of the communications on the TLS channels. In the event that a tampered or modified packet is observed on the channel the TOE will discard the packet and log an entry in the audit log.

7.9 Stateful Traffic Firewall

7.9.1 Overview of Firewall Functionality

The FortiGate™ family appliances are capable of filtering on a large number of parameters, applications and protocols. For the scope of this evaluation this includes filtering on traffic complying with IPv4 (RFC 791), ICMPv4 (RFC 792), TCP (RFC 793), UDP (RFC 768), IPv6 (RFC 2460) and ICMPv6 (RFC 4443). These protocols can be filtered based on source address, destination address, transport layer protocol, type and code attributes of initial packets of any communication session. For TCP and UDP sessions the TOE is capable of inspecting, filtering and logging based on the source and/or destination ports present in the packet headers. For TCP communications only the TOE is capable of inspecting, filtering and logging packets based on the sequence number and flags which are present in the header information. 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. 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.

The TOE also maintains a database of firewall rules. For each rule a number of attributes are defined including interface label on the TOE and assigned a unique order and an action consisting of permit, log or deny. These rules are consulted in their configured order until a match is made with all attributes of the rule (source, destination, protocol, port, code). If there is a match found the corresponding action

of allowing the packet, logging the packet or dropping the packet will be performed. If there is no match found the packet will be denied. Depending on the hardware model these ports may have a variety of name identifiers[13] on them however all ports are treated the same by the underlying operating system and there is no difference on the port label when processing the firewall rules and the names are provided to help administrators locate the correct physical port.

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. Packets that do not match the attributes in the session database are then compared to the defined firewall rules for that interface identifier based on their unique numerical order. Packets that are permitted are passed to their destination, packets marked for logging are written to the audit log and packets marked for dropping are discarded.

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.

7.9.2 Firewall Interoperability

Fortinet has years of experience with interoperability testing and field deployments of their firewalls. This has led to an extensive collection of knowledge-based articles posted on https://kb.fortinet.com. These articles include having the TOE use specific known configuration settings when working with other implementations from other vendors.

The FortiOS™ operating system running on all FortiGate appliances has received IPv6 Ready Logo Program validation from the IPv6 Forum, a worldwide consortium that provides technical guidance for the deployment of IPv6 technology. It has successfully fulfilled all requirements for IPv6 Phase-2 Core Support as a router product, thereby validating the interoperability of FortiGate appliances with other IPv6 products. Additionally the FortiOS operating system has achieved the U.S. Department of Defense (DoD) IPv6 product certification conducted by the Joint Interoperability Test Command (JITC).

7.9.3 Overview of Startup Process

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

[13] These identifiers may be terms such as WiFi, WAN, LAN or DMZ

- kernel
- firmware
- cryptographic known answer tests
- entropy gathering and DRBG initialization
- 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 and the TOE is available for login to accept administrative connections.

7.9.4 Firewall Error Modes

There are several self-tests in which the firewall 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 modes the TOE will halt and no further processing will occur until the TOE is reset.

Additionally the TOE may receive traffic above the capacity of the product it will drop all packets above this capacity. These events are logged to the audit log of the TOE.

7.9.5 Default Stateful Traffic Filtering

By default the TOE will reject packets which match the following rules:

- *packets which are invalid fragments*
- *fragmented IP packets which cannot be re-assembled completely*
- *the source address of the network packet is equal to the address of the network interface where the network packet was received*
- *the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received*
- *the source address of the network packet is defined as being on a broadcast network*
- *the source address of the network packet is defined as being on a multicast network*
- *the source address of the network packet is defined as being a loopback address*
- *the source address of the network packet is a multicast*
- *the source or destination address of the network packet is a link-local address*
- *the source or destination address of the network packet is defined as being an address "reserved for future use" as specified in RFC 5735 for IPv4*
- *the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" as specified in RFC 3513 for IPv6;*
- *the IP options: Loose Source Routing, Strict Source Routing, or Record Route are specified*

The TOE implements Strict Reverse Path Forwarding in order to understand the networks associated with each interface which exists on the TOE.

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 behavior is capable of being modified or overwritten by the TOE administrator.

8 RATIONALE

This ST claims Exact Conformance to Network Devices Protection Profile v1.1 and the NDPP Extended Package Stateful Traffic Filter Firewall v1.0 as well as Errata #2 for the NDPP. Hence, conformance claim rationale, security objectives rationale, extended SFR rationale, and security requirements rationale (including SAR choice rationale) are explicitly addressed by the Protection Profile and the Extended Package, without further elaboration in this ST, with the following exceptions.

The firewall EP talks of a security administrator who is only capable of administering the firewall ruleset for the TOE. This functionality has been grouped together with the "Authorized Administrator" role from the NDPP for the purposes of this document. The TOE is capable of separating and restricting this access, however in the evaluated configuration only one role was implemented.

The dependency rationale is not stated by the NDPP or the FW EP, and a such is provided below.

8.1 Dependency Rationale

Table 11 –Functional Requirements Dependencies

SFR	Dependencies	Dependency Met	Rationale
FAU_GEN.1	FPT_STM.1	Yes	
FAU_GEN.2	FAU_GEN.1	Yes	
	FIA_UID.1	Yes	FIA_UIA_EXT.1 is hierarchical to FIA_UID.1, thus the dependency is satisfied
FAU_STG_EXT.1	FAU_GEN.1	Yes	
	FTP_ITC.1	Yes	
FCS_CKM.1	FCS_CKM.2 or FCS_COP.1	Yes	The TOE implements FCS_COP.1
	FCS_CKM.4	Yes	The ST claims exact conformance to the NDPP which does not have this SFR present. The TOE is FIPS level 1 validated which covers this dependency. The TOE also claims FCS_CKM_EXT.4 is stated to be modeled after FCS_CKM.4 and satisfies the requirement even though it is not stated as hierarchical in the NDPP.
FCS_CKM_EXT.4	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	Yes	The TOE implements FCS_CKM.1
FCS_COP.1(1)	FDP_ITC.1 or FDP_ITC.2 or	Yes	The TOE implements FCS_CKM.1

SFR	Dependencies	Dependency Met	Rationale
	FCS_CKM.1		
	FCS_CKM.4	Yes	The ST claims exact conformance to the NDPP which does not have this SFR present. The TOE is FIPS level 1 validated which covers this dependency. The TOE also claims FCS_CKM_EXT.4 is stated to be modeled after FCS_CKM.4 and satisfies the requirement even though it is not stated as hierarchical in the NDPP.
FCS_COP.1(2)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	Yes	The TOE implements FCS_CKM.1
	FCS_CKM.4	Yes	The ST claims exact conformance to the NDPP which does not have this SFR present. The TOE is FIPS level 1 validated which covers this dependency. The TOE also claims FCS_CKM_EXT.4 is stated to be modeled after FCS_CKM.4 and satisfies the requirement even though it is not stated as hierarchical in the NDPP.
FCS_COP.1(3)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	Yes	The TOE implements FCS_CKM.1
	FCS_CKM.4	Yes	The ST claims exact conformance to the NDPP which does not have this SFR present. The TOE is FIPS level 1 validated which covers this dependency. The TOE also claims FCS_CKM_EXT.4 is stated to be modeled after FCS_CKM.4 and satisfies the requirement even though it is not stated as hierarchical in the NDPP.
FCS_COP.1(4)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	Yes	The TOE implements FCS_CKM.1
	FCS_CKM.4	Yes	The ST claims exact conformance to the NDPP which does not have this SFR present. The TOE is FIPS level 1 validated which covers this dependency.

SFR	Dependencies	Dependency Met	Rationale
			The TOE also claims FCS_CKM_EXT.4 is stated to be modeled after FCS_CKM.4 and satisfies the requirement even though it is not stated as hierarchical in the NDPP.
FCS_HTTPS_EXT.1	FCS_TLS_EXT.1	Yes	
FCS_RBG_EXT.1	None	Yes	
FCS_TLS_EXT.1	FCS_COP.1(1)	Yes	
	FCS_COP.1(2)	Yes	
	FCS_COP.1(3)	Yes	
	FCS_COP.1(4)	Yes	
	FCS_RBG_EXT.1	Yes	
	FCS_CKM.1	Yes	
	FCS_CKM_EXT.4	Yes	
FDP_RIP.2	None	Yes	
FIA_PMG_EXT.1	None	Yes	
FIA_UAU.7	FIA_UAU.1	Yes	FIA_UIA_EXT.1 is hierarchical to FIA_UAU.1, thus the dependency is satisfied
FIA_UAU_EXT.2	None	Yes	
FIA_UIA_EXT.1	FTA_TAB.1	Yes	
FMT_MTD.1	FMT_SMR.1	Yes	
	FMT_SMF.1	Yes	
FMT_SMF.1	None	Yes	
FMT_SMR.2	FIA_UID.1	Yes	FIA_UIA_EXT.1 is hierarchical to FIA_UID.1, thus the dependency is satisfied
FPT_APW_EXT.1	None	Yes	
FPT_SKP_EXT.1	None	Yes	
FPT_STM.1	None	Yes	
FPT_TST_EXT.1	None	Yes	

SFR	Dependencies	Dependency Met	Rationale
FPT_TUD_EXT.1	FCS_COP.1(2) or FCS_COP.1(3)	Yes	The TOE implements signature verification via FCS_COP.1(2)
FTA_SSL.3	None	Yes	
FTA_SSL.4	None	Yes	
FTA_SSL_EXT.1	FIA_UIA_EXT.1	Yes	
FTA_TAB.1	None	Yes	
FTP_ITC.1	None	Yes	
FTP_TRP.1	None	Yes	
FFW_RUL_EXT.1	None	Yes	

9 ACRONYMS

Table 12 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CAVP	Cryptographic Algorithm Validation Program
CC	Common Criteria
CCCS	Canadian Common Criteria Scheme
CEM	Common Evaluation Methodology
CFB	Cipher Feedback
CMVP	Cryptographic Module Validation Program
CSP	Critical Security Parameters
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
FIPS	Federal Information Processing Standard
FSSO	Fortinet Single Sign-On
HMAC	Keyed-Hash Message Authentication Code
NDPP	Network Device Protection Profile
OFB	Output Feedback
OSP	Organizational Security Policy
PP	Protection Profile
RBG	Random Bit Generator
SA	Security Association
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSL	Secure Socket Layer
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality

10 APPENDIX A – HARDWARE PLATFORM DETAILS

10.1 Desktop Form Factor

Model	CPU	ASIC	RAM	Flash	Storage
FG-20C	ARM v5 Compatible (SoC)	CP6	512MB	2GB	N/A
FWF-20C	ARM v5 Compatible (SoC)	CP6	512MB	2GB	N/A
FG-30D	ARM v5 Compatible (SoC2)	CP7	1GB	4GB	N/A
FWF-30D	ARM v5 Compatible (SoC2)	CP7	1GB	4GB	N/A
FWF-30D-PoE	ARM v5 Compatible (SoC2)	CP7	1GB	4GB	N/A
FG-40C	ARM v5 Compatible (SoC)	CP6	512MB	2GB	N/A
FWF-40C	ARM v5 Compatible (SoC)	CP6	512MB	2GB	N/A
FG-60C	ARM v5 Compatible (SoC)	CP6	512MB	8GB	8GB
FG-60D	ARM v5 Compatible (SoC2)	CP7	2GB	8GB	8GB
FG-60D-PoE	ARM v5 Compatible (SoC2)	CP7	2GB	8GB	8GB
FWF-60D	ARM v5 Compatible (SoC2)	CP7	2GB	8GB	8GB
FG-80C	Intel Tolapai	CP6	512MB	512MB	N/A
FWF-80CM	Intel Celeron	CP6	512MB	512MB	N/A
FG-90D	ARM v5 Compatible (SoC2)	CP7	2GB	2GB	32GB
FG-90D-PoE	ARM v5 Compatible (SoC2)	CP7	2GB	8GB	32GB
FG-110C	Intel Celeron	CP6	1GB	64MB	N/A
FG-111C	Intel Celeron	CP6	1GB	64MB	N/A

10.2 1U Form Factor

Model	CPU	ASIC	RAM	Flash	Storage
FG-100D	Intel Atom	CP8	2GB	16GB	32GB
FG-140D	Intel Atom	CP8	4GB	16GB	32GB
FG-140D-PoE	Intel Atom	CP8	4GB	16GB	32GB

FG-200B	Intel Celeron	CP6	1GB	4GB	64GB
FG-200B-PoE	Intel Celeron	CP6	1GB	4GB	64GB
FG-200D	Intel G540	CP8	4GB	16GB	64GB
FG-240D	Intel Celeron	CP8	4GB	4GB	64GB
FG-300C	Intel Celeron	CP6	2GB	16GB	16GB
FG-300D	Intel i3-3220	CP8	8GB	16GB	120GB
FG-310B	Intel Celeron	CP6	1GB	128MB	N/A
FG-311B	Intel Celeron	CP6	1GB	128MB	N/A
FG-500D	Intel Xeon E Series	CP8	8GB	16GB	120GB
FG-600C	Intel i3-540	CP8	4GB	8GB	64GB
FG-620B	Intel Core 2 Duo	CP6	2GB	512MB	80GB (AMC module)
FG-621B	Intel Core 2 Duo	CP6	2GB	512MB	80GB (AMC module)
FG-800C	Intel i5-750	CP8	8GB	8GB	64GB

10.3 2U Form Factor

Model	CPU	ASIC	RAM	Flash	Storage
FG-280D-PoE	Intel Celeron	CP8	4GB	4GB	64GB
FG-1000C	Intel i5 Quad Core	CP8	8GB	8GB	128GB
FG-1000D	Intel Xeon E Series	CP8	16GB	4GB	256GB
FG-1200D	Intel Xeon E Series	CP8	16GB	16GB	240GB
FG-1240B	Intel i5-750	CP8	8GB	8GB	128GB
FG-1500D	Intel Xeon E Series	CP8	16GB	32GB	480GB
FG-3040B	Intel Xeon E Series	CP7	12GB	8GB	64GB
FG-3140B	Intel Xeon E Series	CP7	12GB	8GB	64GB
FG-3240C	Intel Xeon E5 Series	CP8	12GB	8GB	64GB

10.4 3U Form Factor

Model	CPU	ASIC	RAM	Flash	Storage
FG-3600C	Intel Xeon E Series	CP8	32GB	2GB	64GB
FG-3700D	Intel Xeon E Series	CP8	64GB	16GB	960GB
FG-3950B	Intel Xeon E Series	CP7	12GB	8GB	N/A
FG-3951B	Intel Xeon E Series	CP7	12GB	8GB	N/A

10.5 Blade Form Factor

Model	CPU	ASIC	RAM	Flash	Storage
FG-5001A	Intel Xeon E5 Series	CP6	4GB	128MB	N/A
FG-5001B	Intel Xeon LC Series	CP7	12GB	8GB	64GB
FG-5001C	Intel Xeon E5-2658L	CP8	32GB	32GB	128GB
FG-5001D	Intel Xeon E Series	CP8	32GB	16GB	200GB
FG-5101C	Intel Xeon LC Series	CP8	12GB	8GB	64GB
FSW-5203B	Intel Xeon 3500 Series	CP8	12GB	8GB	64GB