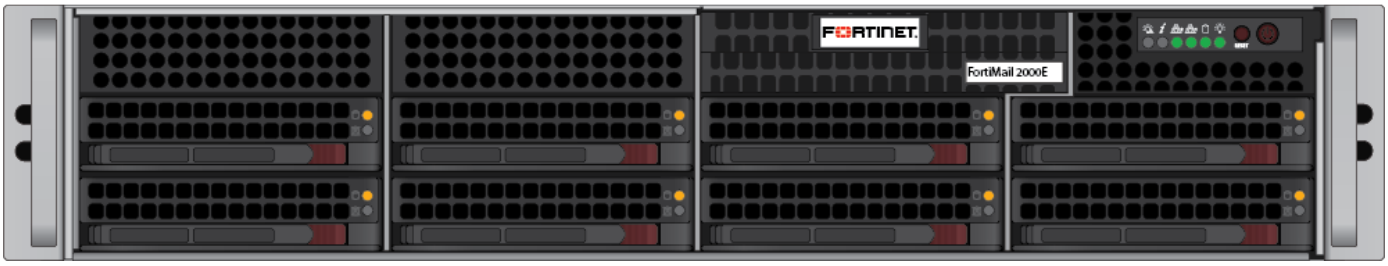
FIPS 140-2 Non-Proprietary Security Policy

FortiMail 6.0



FortiMail 6.0 FIPS 140-2 Security Policy	
Document Version:	1.7
Publication Date:	February 8, 2019
Description:	Documents FIPS 140-2 Level 2 Security Policy issues, compliancy and requirements for FIPS compliant operation.
Firmware Version:	FortiMail v6.0, build108, 180731

FORTINET DOCUMENT LIBRARY

https://docs.fortinet.com

FORTINET VIDEO GUIDE

https://video.fortinet.com

FORTINET KNOWLEDGE BASE

http://kb.fortinet.com

FORTINET BLOG

https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT

https://support.fortinet.com

FORTINET COOKBOOK

http://cookbook.fortinet.com

FORTINET NSE INSTITUTE (TRAINING)

https://training.fortinet.com/

FORTIGUARD CENTER

https://fortiguard.com

FORTICAST

http://forticast.fortinet.com

END USER LICENSE AGREEMENT AND PRIVACY POLICY

https://www.fortinet.com/doc/legal/EULA.pdf

https://www.fortinet.com/corporate/about-us/privacy.html

FEEDBACK

Email: techdocs@fortinet.com

Friday, February 8, 2019

FortiMail 6.0 FIPS 140-2 Non-Proprietary Security Policy

06-602-508667-20180817

TABLE OF CONTENTS

Overview **4**
References 4
Introduction **5**
Security Level Summary **6**
Module Descriptions **7**
Module Interfaces 8
Web-Based Manager 10
Command Line Interface 10
Roles, Services and Authentication 11
Roles 11
FIPS Approved Services 11
Non-FIPS Approved Services 13
Authentication 13
Operational Environment 14
Cryptographic Key Management 14
Random Number Generation 14
Entropy 14
Key Zeroization 15
Algorithms 15
Cryptographic Keys and Critical Security Parameters 17
Alternating Bypass Feature 20
Key Archiving 20
Mitigation of Other Attacks 20
FIPS 140-2 Compliant Operation **21**
Enabling FIPS-CC mode 22
Self-Tests **23**
Startup and Initialization Self-tests 23
Conditional Self-tests 23
Critical Function Self-tests 24
Error State 24

Overview

This document is a FIPS 140-2 Security Policy for Fortinet's FortiMail 6.0 firmware, which runs on the FortiMail family of security appliances. This policy describes how the FortiMail 6.0 firmware (hereafter referred to as the 'module') meets the FIPS 140-2 security requirements and how to operate the module in a FIPS compliant manner. This policy was created as part of the FIPS 140-2 Level 2 validation of the module.

The Federal Information Processing Standards Publication 140-2 - *Security Requirements for Cryptographic Modules* (FIPS 140-2) details the United States Federal Government requirements for cryptographic modules. Detailed information about the FIPS 140-2 standard and validation program is available on the NIST (National Institute of Standards and Technology) website at http://csrc.nist.gov/groups/STM/cmvp/index.html.

References

This policy deals specifically with operation and implementation of the modules in the technical terms of the FIPS 140-2 standard and the associated validation program. Other Fortinet product manuals, guides and technical notes can be found at the Fortinet technical documentation website at http://docs.fortinet.com.

Additional information on the entire Fortinet product line can be obtained from the following sources:

- Find general product information in the product section of the Fortinet corporate website at http://www.fortinet.com/products.
- Find on-line product support for registered products in the technical support section of the Fortinet corporate website at http://www.fortinet.com/support.
- Find contact information for technical or sales related questions in the contacts section of the Fortinet corporate website at http://www.fortinet.com/contact.
- Find security information and bulletins in the FortiGuard Center of the Fortinet corporate website at http://fortiguard.com.

Introduction

The FortiMail family of message security appliances provide an effective barrier against the ever-rising volume of spam, maximum protection against sophisticated messagebased attacks, and features designed to facilitate regulatory compliance. FortiMail 5.4 offers both inbound and outbound scanning, advanced antispam and antivirus filtering capabilities, IP address black/white listing functionality, and extensive quarantine and archiving capabilities. Three deployment modes offer maximum versatility: transparent mode for seamless integration into existing networks with no IP address changes, gateway mode as a proxy Mail Transfer Agent (MTA) for existing messaging gateways, or server mode to act as a mail server with functionality for small businesses (SMBs) and remote offices.

Note: The server mode of operation is not a FIPS approved mode of operation.

Security Level Summary

The module meets the overall requirements for a FIPS 140-2 Level 2 validation.

Table 1: Summary of FIPS security requirements and compliance levels

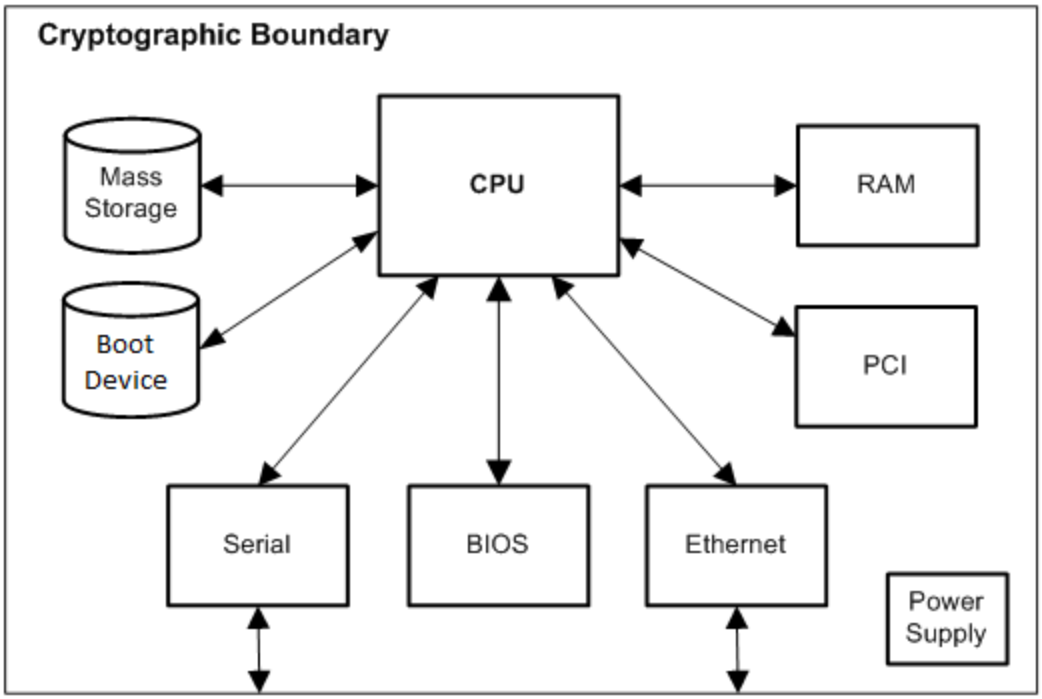
Security Requirement	Compliance Level
Cryptographic Module Specification	2
Cryptographic Module Ports and Interfaces	2
Roles, Services and Authentication	3
Finite State Model	2
Physical Security	2
Operational Environment	N/A
Cryptographic Key Management	2
EMI/EMC	2
Self-Tests	2
Design Assurance	3
Mitigation of Other Attacks	N/A

Module Descriptions

The module is a firmware operating system that runs exclusively on Fortinet's FortiMail product family. FortiMail units are PC-based, purpose built appliances.

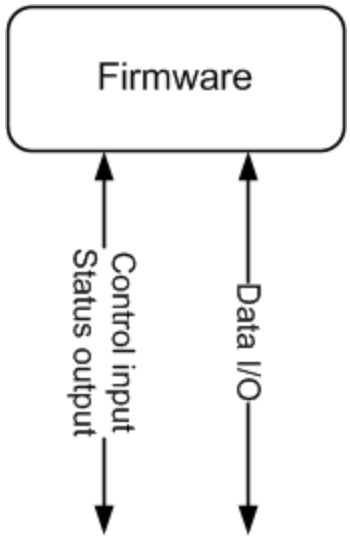
The FortiMail appliances are multiple chip, standalone cryptographic modules consisting of production grade components contained in a physically protected enclosure.

Figure 1 - FortiMail physical cryptographic boundary



The Boot Device in the diagram above can refer to a separate, internal component or a partition on the Mass Storage device. All references herein of 'boot device' shall refer to the configuration specific to the FortiMail appliance.

Figure 2 - FortiMail logical cryptographic boundary



For the purposes of FIPS 140-2 conformance testing, the module was tested on a FortiMail-2000E appliance and used a Fortinet entropy token (FTR-ENT-1) as the entropy source.

The validated firmware version is FortiMail v6.0, build108, 180731. Any firmware version that is not shown on the module certificate is out of scope of this validation and requires a separate FIPS 140-2 validation.

The module can also be executed on any of the following FortiMail appliances and remain vendor affirmed FIPS-compliant. As per IG G.5, the recompilation per appliance does not require any source code modifications.

Table 2: Vendor affirmed FIPS-compliance appliances

FortiMail-60D	FortiMail-1000D
FortiMail-200D	FortiMail-3000D
FortiMail-200E	FortiMail-3000E
FortiMail-400E	FortiMail-3200E
FortiMail-400F	

Module Interfaces

The module's logical interfaces and physical ports are described in the table below.

Table 3: FortiMail logical interfaces and physical ports

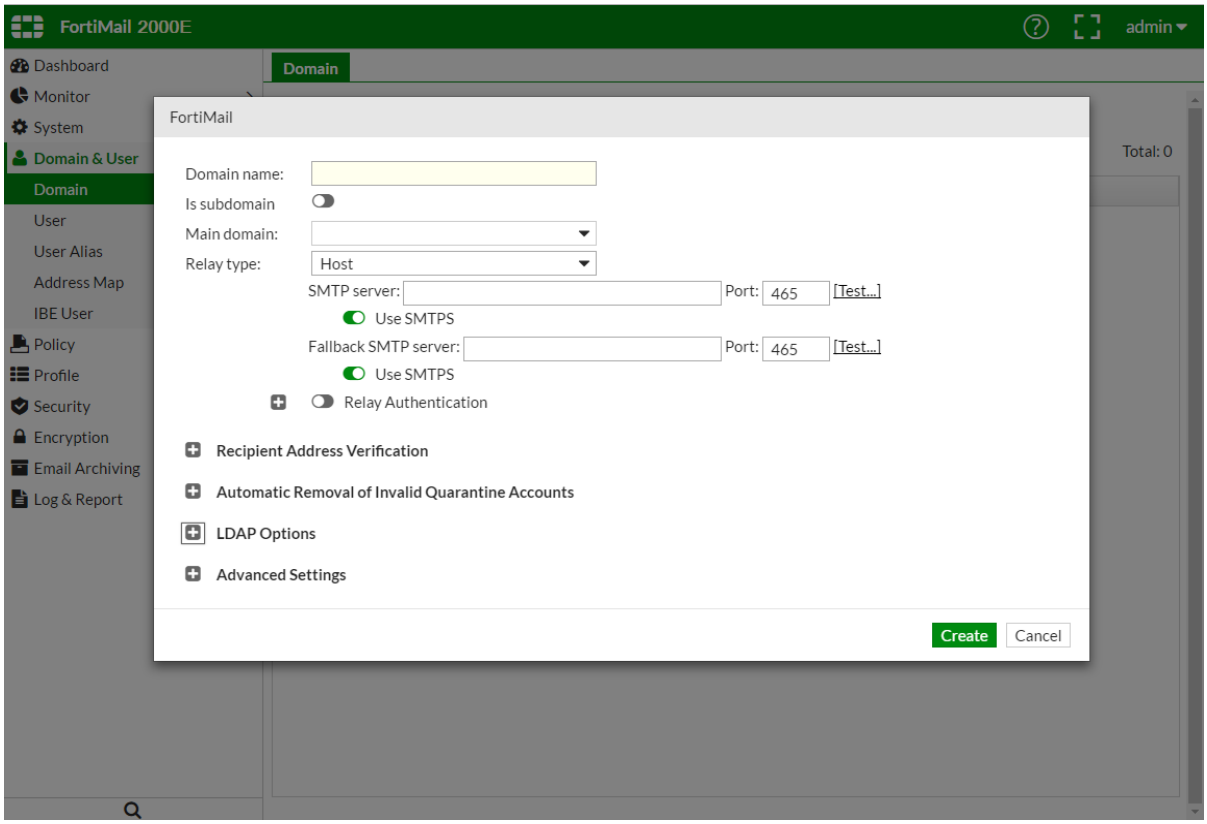
FIPS 140 Interface	Logical Interface	Physical Interface
Data Input	API input parameters	Network interface, USB interface (Entropy Token)
Data Output	API output parameters	Network Interface
Control Input	API function calls	Network Interface, serial interface, USB interface (USB token)
Status Output	API return values	Network interface, serial interface
Power Input	n/a	The power supply is the power interface

Web-Based Manager

The FortiMaill web-based manager provides GUI based access to the modules and is the primary tool for configuring the modules. The manager requires a web browser on the management computer and an Ethernet connection between the FortiMail unit and the management computer.

A web-browser that supports Transport Layer Security (TLS) 1.1 or 1.2 is required for remote access to the web-based manager when the module is operating in FIPS-CC mode. HTTP access to the web-based manager is not allowed in FIPS mode and is disabled.

Figure 3 - The FortiMail web based manager



Command Line Interface

The FortiMail Command Line Interface (CLI) is a full-featured, text based management tool for the module. The CLI provides access to all of the possible services and configuration options in the module. The CLI uses a console connection or a network (Ethernet) connection between the FortiMail unit and the management computer. The console connection is a direct serial connection. Terminal emulation software is required on the management computer using either method. For network access, a Telnet or SSH client that supports the SSH

v2.0 protocol is required (SSH v1.0 is not supported in FIPS mode). Telnet access to the CLI is not allowed in FIPS mode and is disabled.

Roles, Services and Authentication

Roles

When configured in FIPS mode, the module provides the following roles:

- Crypto Officer
- User

The Crypto Officer role is initially assigned to the default 'admin' operator account. The Crypto Officer role has read-write access to all of the module's administrative services. The initial Crypto Officer can create additional operator accounts. These additional accounts are assigned the Crypto Officer role and can be assigned a range of read/write or read only access permissions including the ability to create operator accounts.

The User role can make use of the encrypt/decrypt services, but cannot access the module for administrative purposes. The User role has access to the quarantine and email relay services as defined by a Crypto Officer.

Operators can be logged in concurrently and separation is enforced via separate identities.

The module does not provide a Maintenance role.

FIPS Approved Services

The following tables detail the types of FIPS approved services available to each role in each mode of operation, the types of access for each role and the Keys or CSPs they affect.

The access types are abbreviated as follows:

Read Access	R
Write Access	W
Execute Access	E

Table 4: Services available to Crypto Officers

Service	Access	Key/CSP
connect to module locally using the console port	WE	N/A

Service	Access	Key/CSP
connect to module remotely using TLS*	WE	Diffie-Hellman Key, EC Diffie Hellman Key, HTTPS/TLS Premaster Secret and Master Secret, HTTPS/TLS Server/Host Key, HTTPS/TLS Session Authentication Key, and HTTPS/TLS Session Encryption Key, DRBG v and key values, DRBG Output, DRBG Seed, NDRNG Output String
connect to module remotely using SSH*	WE	Diffie-Hellman Key, SSH Server/Host Key, SSH Session Authentication Key, SSH Session Encryption Key, DRBG v and key values, DRBG Output, DRBG Seed, NDRNG Output String
authenticate to module	WE	Crypto Officer Password
show system status	WE	N/A
show FIPS-CC mode enabled/disabled (console/CLI only)	WE	N/A
enable FIPS-CC mode of operation (console only)	WE	Configuration Integrity Key
key zeroization	WE	All Keys
execute factory reset (disable FIPS-CC mode, console/CLI only)	E	All keys stored in Flash RAM
execute FIPS-CC on-demand self-tests (console only)	E	Configuration Integrity Key, Firmware Integrity Key
add/delete crypto officer and users	WE	Crypto Officer Password,User Password
set/reset crypto officer and user passwords	WE	Crypto Officer Password, User Password
modify user preferences	RWE	N/A
backup/restore configuration file	WE	Configuration Encryption Key, Configuration Backup Key
read/set/delete/modify module configuration	RWE	N/A

Service	Access	Key/CSP
execute firmware update	E	Firmware Update Key
read log data	R	N/A
delete log data (console/CLI only)	WE	N/A
format log disk (console/CLI only)	WE	N/A
enable/disable alternating bypass mode	WE	N/A
read/set/modify HA configuration	WE	HA Password, HA Encryption Key

Table 5: Services available to Users in FIPS-CC mode

Service/CSP	Access	Key/CSP
connect to module remotely using TLS*	WE	Diffie-Hellman Keys, EC Diffie-Hellman Keys, HTTPS/TLS Premaster Secret and Master Secret, HTTPS/TLS Server/Host Key, HTTPS/TLS Session Authentication Key, HTTPS/TLS Session Encryption Key, DRBG v and key values, DRBG Output, DRBG Seed, NDRNG Output String
authenticate to module	WE	User Password
access to quarantined email	RE	N/A
modify user preferences	E	N/A

Non-FIPS Approved Services

The module also provides the following non-FIPS approved services:

- Configuration backups using password protection
- Services marked with an asterisk (*) in Tables 4 and 5 are considered non-approved when using the following algorithms:
    - Non-compliant-strength Diffie-Hellman
    - Non-compliant-strength RSA key wrapping

The above services shall not be used in the FIPS approved mode of operation.

Authentication

The module uses identity based authentiction. By default, operators and users authenticate with a username and password combination to access the module. The username/password can be stored in the local database or in a remote LDAP database. Remote operator authentication is done over HTTPS (TLS) or SSH. Local operator

authentication is done over the console connection. Remote user authentication is done over HTTPS (TLS). Password entry is obfuscated using asterisks.

Note that operator authentication over HTTPS/SSH and user authentication over HTTPS are subject to a limit of 3 failed authentication attempts in 1 minute; thus, the maximum number of attempts in one minute is 3. Therefore the probability of a success with multiple consecutive attempts in a one-minute period is 3 in {(10)*(26^2)*(32)*(94^4)} which is less than 1/100,000. Operator authentication using the console is not subject to a failed authentication limit, but the number of authentication attempts per minute is limited by the bandwidth available over the serial connection which is a maximum of 115,200 bps which is 6,912,000 bits per minute. An 8 byte password would have 64 bits, so there would be no more than 108,000 passwords attempts per minute. Therefore the probability of success would be 1/({(10)*(26^2)*(32)*(94^4)} /108,000) which is less than 1/100,000.

Note that the user's username and password are not stored on the module. The module operates as a proxy for user authentication to a backend server (typically a mail server). User authentication is done over HTTPS, POP3S, or IMAPS. HTTPS, POP3S and IMAPS all use the underlying TLS protocol to protect user data between the client and the module and the module and the back end server during the authentication process.

The minimum password length is 8 characters when in FIPS-CC mode (maximum password length is 32 characters) chosen from the set of ninety four (94) characters. New passwords are required to include 1 uppercase character, 1 lowercase character, 1 numeric character, and 1 special character. The odds of guessing a password are 1 in {(10)*(26^2)*(32)*(94^4)} which is significantly lower than one in a million.

Operational Environment

The module constitutes the entire firmware operating system for a FortiMail unit and can only be installed and run on a FortiMail unit. The module provides a proprietary and non-modifiable operating system and does not provide a programming environment.

For the purposes of FIPS 140-2 conformance testing, the module was tested on a FortiMail-2000E unit.

Cryptographic Key Management

Random Number Generation

The modules use a firmware based, deterministic random bit generator (DRBG) that conforms to NIST Special Publication 800-90A.

Entropy

The module uses a Fortinet entropy token (part number FTR-ENT-1 or part number FTR-ENT-2) to seed the DRBG during the modules' boot process and to periodically reseed the DRBG. The entropy token is not included in the boundary of the module and therefore no assurance can be made for the correct operation of the entropy token nor is there a guarantee of stated entropy.

Entropy Strength

The entropy loaded into the approved AES-256 bit DRBG is 256 bits. The entropy source is over-seeded and then an HMAC-SHA-256 post-conditioning component is applied.

Reseed Period

The RBG is seeded from the entropy token during the boot process and then reseeded periodically. The default reseed period is once every 24 hours (1440 minutes) and is configurable (1 to 1440 minutes). The entropy token must be installed to complete the boot process and to reseed the DRBG.

Key Zeroization

The zeroization process must be performed under the direct control of the operator. The operator must be present to observe that the zeroization method has completed successfully.

All keys and CSPs are zeroized by erasing the module's boot device and then power cycling the FortiMail unit. To erase the boot device, execute the following command from the CLI:

```
execute erase-disk <boot device>
```

The boot device ID may vary depending on the FortiMail module. Executing the following command will output a list of the available internal disks:

```
execute erase-disk ?
```

Algorithms

Table 6: FIPS approved algorithms

Algorithm	NIST Certificate Number
AES in CBC mode (128-, 256-bits)	5321
AES in GCM mode (128-, 256-bits)	5321
CTR DRBG (NIST SP 800-90A) with 256-bits	2050
CVL (SSH) - AES 128 bit-, AES 256 bit -CBC (using SHA-256)	1787
CVL (TLS 1.1 and 1.2)	1787
CVL(KAS) • FFC "dhEphem" (FC: SHA: SHA-256) • ECC "Ephemeral Unified" (EC: P-256 and ED: P-384)	1786
HMAC SHA-1	3517
HMAC SHA-256	3517
HMAC SHA-384	3517

Algorithm	NIST Certificate Number
RSA • Key Pair Generation: 2048 and 3072-bit (FIPS 186-4) • Signature Generation: 2048 and 3072-bit (PKCS1 v1.5) • Signature Verification: 1024, 2048 and 3072-bit (PKCS1 v1.5) • For legacy use, the module supports 1024-bit RSA keys and SHA-1 for signature verification	2849
SHA-1	4271
SHA-256	4271
SHA-384	4271

KTS (AES Cert. #5321 and HMAC Cert. #3517; key establishment methodology provides 128 or 256 bits of encryption strength).

In accordance with FIPS 140-2 IG D.12, the cryptographic module performs Cryptographic Key Generation (CKG) as per SP800-133 (vendor affirmed). The resulting generated symmetric key and the seed used in the asymmetric key generation are the unmodified output from SP800-90A DRBG.

There are algorithms, modes, and keys that have been CAVs tested but are not available when the module is configured for FIPS compliant operation. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in this table are supported by the module in the FIPS validated configuration.

Table 7: FIPS allowed algorithms

Algorithm
Diffie-Hellman (CVL Certs. #1786 and Cert. #1787, key agreement; key establishment methodology provides between 112 and 201 bits of encryption strength)
EC Diffie-Hellman (CVL Certs. #1786 and Cert. #1787, key agreement; key establishment methodology provides between 128 and 256 bits of encryption strength)
MD5 (used in the TLS protocol only)
NDRNG (Entropy Token)
RSA (key wrapping; key establishment methodology provides 112 or 128 bits of encryption strength)

Table 8: Non-FIPS approved algorithms

Algorithm
Diffie-Hellman is non-compliant when keys less than 2048 bits are used, since such keys do not provide the minimum required 112 bits of encryption strength.

Algorithm
RSA is non-compliant when keys less than 2048 bits are used, since such keys do not provide the minimum required 112 bits of encryption strength.

Note that the SSH and TLS protocols, other than the KDF, have not been tested by the CMVP or CAVP as per FIPS 140-2 Implementation Guidance D.11.

The module is compliant to IG A.5: GCM is used in the context of TLS.

For TLS, The GCM implementation meets Option 1 of IG A.5: it is used in a manner compliant with SP 800-52 and in accordance with RFC 5246 for TLS key establishment. The AES GCM IV generation is in compliance with RFC 5288 and shall only be used for the TLS protocol version 1.2 to be compliant with FIPS140-2 IG A.5, Option 1 (“TLS protocol IV generation”); thus, the module is compliant with [SP800-52]. During operational testing, the module was tested against an independent version of TLS and found to behave correctly.

In case the module’s power is lost and then restored, the key used for the AES GCM encryption or decryption shall be re-distributed.

Cryptographic Keys and Critical Security Parameters

The following table lists all of the cryptographic keys and critical security parameters used by the modules.

Table 9: Cryptographic Keys and Critical Security Parameters used in FIPS-CC mode

Key or CSP	Generation	Storage	Usage	Zeroization
NDRNG output string	Automatic	Boot device Plain-text	Input string for the entropy pool (5120-bits)	By erasing the Boot device and power cycling the module
DRBG seed	Automatic	Boot device Plain-text	256-bit seed used by the DRBG (output from NDRNG	By erasing the Boot device and power cycling the module
DRBG output	Automatic	Boot device Plain-text	Random numbers used in cryptographic algorithms (256-bits)	By erasing the Boot device and power cycling the module
DRBG v and key values	Automatic	Boot device Plain-text	Internal state values for the DRBG	By erasing the Boot device and power cycling the module
Diffie-Hellman Keys	Automatic	SDRAM Plain-text	Key agreement and key establishment	By erasing the boot device and power cycling the module

Key or CSP	Generation	Storage	Usage	Zeroization
EC Diffie-Hellman Keys	Automatic	SDRAM Plain-text	Key agreement and key establishment (key pairs on the curves secp256r1, secp384r1 and secp521r1	By erasing the boot device and power cycling the module
Firmware Update Key	Preconfigured	Boot device Plain-text	Verification of firmware integrity when updating to new firmware versions using RSA public key (firmware load test, 2048-bit signature)	By erasing the boot device and power cycling the module
Firmware Integrity Key	Preconfigured	Boot device Plain-text	Verification of firmware integrity in the firmware integrity test using RSA public key (firmware integrity test, 2048-bit signature)	By erasing the boot device and power cycling the module
HTTPS/TLS Server/Host Key	Preconfigured	Boot device Plain-text	RSA private key used in the HTTPS/TLS protocols (key establishment, 2048- or 3072-bit)	By erasing the boot device and power cycling the module
HTTPS/TLS Pre-Master Secret	Automatic	SDRAM Plain-text	Generation of HTTPS/TLS Master Secret (384 bits)	By erasing the boot device and power cycling the module
HTTP/TLS Master Secret	Automatic	SDRAM Plain-text	Generation of TLS Session Keys and TLS Authentication Key (384-bits)	By erasing the boot device and power cycling the module
HTTPS/TLS Session Authentication Key	Automatic	SDRAM Plain-text	HMAC SHA-1, -256 or -384 key used for HTTPS/TLS session authentication	By erasing the boot device and power cycling the module
HTTPS/TLS Session Encryption Key	Automatic	SDRAM Plain-text	AES key used for HTTPS/TLS session encryption	By erasing the boot device and power cycling the module

Key or CSP	Generation	Storage	Usage	Zeroization
SSH Server/Host Key	Preconfigured	Boot device Plain-text	RSA private key used in the SSH protocol (key establishment, 2048- or 3072-bit)	By erasing the boot device and power cycling the module
SSH Session Authentication Key	Automatic	SDRAM Plain-text	HMAC SHA-1 or HMAC SHA-256 key used for SSH session authentication	By erasing the boot device and power cycling the module
SSH Session Encryption Key	Automatic	SDRAM Plain-text	AES (128-, 256-bit) key used for SSH session encryption	By erasing the boot device and power cycling the module
Crypto Officer Password	Manual	Boot device SHA-1 hash	Used to authenticate operator access to the module	By erasing the boot device and power cycling the module
Configuration Integrity Key	Preconfigured	Boot device Plain-text	HMAC SHA-256 hash used for configuration integrity test	By erasing the boot device and power cycling the module
Configuration Encryption Key	Preconfigured	Boot device Plain-text	AES 256-bit key used to encrypt CSPs on the Boot device and in the backup configuration file (except for crypto officer passwords in the backup configuration file)	By erasing the boot device and power cycling the module
Configuration Backup Key	Automatic	Boot device Plain-text	HMAC SHA-256 key used to hash crypto officer passwords in the backup configuration file	By erasing the boot device and power cycling the unit
User Password	Manual	Boot device SHA-256 hash	Used to authenticate network access to the module	By erasing the boot device and power cycling the unit
HA Password	Manual	Boot device AES encrypted	Used to authenticate FortiMail units in an HA cluster	By erasing the boot device and power cycling the unit
HA Encryption Key	Manual	Boot device AES encrypted	Encryption of traffic between units in an HA cluster using AES 128-bit key	By erasing the boot device and power cycling the unit

The Generation column lists all of the keys/CSPs and their entry/generation methods. Manual entered keys are entered by the operator electronically (as defined by FIPS) using the console or a management computer. Pre-configured keys are set as part of the firmware (hardcoded) and are not operator modifiable. Automatic keys are generated as part of the associated protocol.

Alternating Bypass Feature

The primary cryptographic function of the module is encrypting/decrypting email messages sent/received using SMTP over TLS (SMTPS). The module can also send/receive plain-text email messages using SMTP. The module implements an alternating bypass feature based on the module's configuration and the direction of traffic. If the traffic is sent/received using SMTPS, the module is operating in a non-bypass state. If the traffic is sent/received using SMTP, the module is operating in a bypass state.

Incoming traffic is processed according to the protocol used and the domain configuration. An SMTPS message received by the module is decrypted before being processed. Once processed, if the specified domain is configured to use SMTPS, the message is encrypted before being sent to the mail server (non-bypass state). If the specified domain is configured to use SMTP, then the message is sent to the mail server in plain-text (bypass state).

Outgoing traffic is processed according to the message delivery configuration. If the destination domain is configured to use SMTPS, then the message is encrypted before it is sent (non-bypass state). If the destination domain is configured to use SMTP, then the message is sent in plain-text (bypass state).

Use of SMTPS for incoming traffic is enabled/disabled by checking/unchecking the "Use SMTPS" checkbox in the domain configuration. Use of SMTPS for outgoing traffic is enabled/disabled by creating a delivery policy with valid TLS and encryption profiles.

Key Archiving

The module supports key archiving to a management computer as part of the module configuration file backup. Operator entered keys are archived as part of the module configuration file. The configuration file is stored in plain text, but keys in the configuration file are either AES encrypted using the Configuration Encryption Key or stored as a keyed hash using HMAC SHA-256 using the Configuration Backup Key.

Mitigation of Other Attacks

The module does not mitigate against any other attacks.

FIPS 140-2 Compliant Operation

The Fortinet hardware is shipped in a non-FIPS 140-2 compliant configuration. The following steps must be performed to put the module into a FIPS compliant configuration:

1. Download the model specific FIPS validated firmware image from the Fortinet Support site at https://support.fortinet.com/
2. Verify the integrity of the firmware image
3. Install the FIPS validated firmware image
4. Install the entropy token
5. Enable the FIPS-CC mode of operation

These steps are described in detail in the "add technote link" document that can be found on the Fortinet Technical Documentation website.

In addition, FIPS 140-2 compliant operation requires both that you use the module in its FIPS-CC mode of operation and that you follow secure procedures for installation and operation of the FortiMail unit. You must ensure that:

- The FortiMail unit is configured in the FIPS-CC mode of operation.
- The FortiMail unit is installed in a secure physical location.
- The tamper seals are applied as per the Physical Security instructions.
- Physical access to the FortiMail unit is restricted to authorized operators.
- The Fortinet entropy token is enabled.
- The Fortinet entropy token remains in the USB port during operation.
- Administrative passwords are at least 8 characters long.
- Administrative passwords are changed regularly.
- Administrator account passwords must have the following characteristics:
  - One (or more) characters must be capitalized
  - One (or more) chartacters must be lower case
  - One (or more) characters must be numeric
  - One (or more) characters must be non alpha-numeric (e.g. punctuation mark)
- Administration of the module is permitted using only validated administrative methods. These are:
  - Console connection
  - Web-based manager via HTTPS
  - Command line interface (CLI) access via SSH
- Diffie-Hellman groups of less than 2048 bits are not used.
- Client side RSA certificates must use 2048 bit or greater key sizes.
- Only approved and allowed algorithms are used.

The module can be used in either the Gateway or Transparent modes of operation as described in the Introduction. Note that "mode of operation" in this context does not refer or have any impact on the FIPS approved mode of operation. The FIPS approved mode of operation is independent of the Gateway and Transparent modes of operation. The current operation mode is displayed on the web-based manager status page and in the output of the `get system status` CLI command.

Once the FIPS validated firmware has been installed and the module properly configured in the FIPS-CC mode of operation, the module is running in a FIPS compliant configuration. It is the responsibility of the CO to ensure the module only uses approved algorithms and services to maintain the module in a FIPS-CC Approved mode of operation. Using any of the non-approved algorithms and services switches the module to a non-FIPS mode of operation. Prior to switching between modes the CO should ensure all keys and CSPs are zeroized to prevent sharing of keys and CSPs between the FIPS Approved and non-FIPS mode of operation.

Enabling FIPS-CC mode

To enable the FIPS 140-2 compliant mode of operation, the operator must execute the following command from the Local Console:

```
config system fips-cc
   set status enable
end
```

The Operator is required to supply a password for the admin account which will be assigned to the Crypto Officer role. The supplied password must be at least 8 characters long and correctly verified before the system will restart in FIPS-CC mode. Upon restart, the module will execute self-tests to ensure the correct initialization of the module's cryptographic functions.

After restarting, the Crypto Officer can confirm that the module is running in FIPS-CC mode by executing the following command from the CLI:

```
get system status
```

If the module is running in FIPS-CC mode, the system status output will display the line:

```
FIPS-CC mode: enable
```

Self-Tests

Startup and Initialization Self-tests

The module executes the following self-tests during startup and initialization:

- Firmware integrity test using RSA 2048-bit signatures
- Configuration bypass test using HMAC SHA-256
- AES, CBC mode, encrypt known answer test
- AES, CBC mode, decrypt known answer test
- AES, GCM mode, encrypt known answer test
- AES, GCM mode, decrypt known answer test
- HMAC SHA-1 known answer test
- SHA-1 known answer test (tested as part of HMAC SHA-1 known answer test)
- HMAC SHA-256 known answer test
- SHA-256 known answer test (tested as part of HMAC SHA-256 known answer test)
- HMAC SHA-384 known answer test
- SHA-384 known answer test (tested as part of HMAC SHA-384 known answer test
- RSA signature generation known answer test
- RSA signature verification known answer test
- DRBG known answer test

The results of the startup self-tests are displayed on the console during the startup process.

The startup self-tests can also be initiated on demand using the CLI command **`execute fips kat all`**(to initiate all self-tests) or **`execute fips kat <test>`** (to initiate a specific self-test).

When the self-tests are run, each implementation of an algorithm is tested - i.e. when the AES self-test is run, all AES implementations are tested.

Conditional Self-tests

The module executes the following conditional tests when the related service is invoked:

- Continuous NDRNG test
- Continuous DRBG test
- RSA pairwise consistency test
- Configuration integrity test using HMAC SHA-256
- Firmware load test using RSA signatures

Critical Function Self-tests

The module also performs the following critical function self-tests applicable to the DRBG, as per NIST SP 800-90A Section 11:

- Instantiate test
- Generate test
- Reseed test
- Uninstantiate test

Error State

If any of the self-tests or conditional tests fail, the module enters an error state as shown by the console output below:

```
Self-tests failed
Entering error mode...
The system is going down NOW !!
The system is halted.
```

All data output and cryptographic services are inhibited in the error state.