

FIPS 140-2 Non-Proprietary Security Policy

FortiGate-VM Virtual Appliances

FortiGate-VM Virtual Appliances FIPS 140-2 Security Policy	
Document Version:	2.5
Publication Date:	Tuesday, September 3, 2019
Description:	Documents FIPS 140-2 Level 1 Security Policy issues, compliancy and requirements for FIPS compliant operation.
Software Version:	FortiGate-VM 5.4, b3276, 171006

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 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: techdoc@fortinet.com



Tuesday, September 3, 2019

FortiOS 5.4 FIPS 140-2 Non-Proprietary Security Policy

01-544-418199-20170425

This document may be freely reproduced and distributed whole and intact when including the copyright notice found on the last page of this document.

TABLE OF CONTENTS

Overview	4
References.....	4
Introduction	5
Security Level Summary	6
Module Descriptions	7
Module Interfaces.....	9
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.....	14
Operational Environment.....	14
Cryptographic Key Management.....	15
Random Number Generation.....	15
Entropy.....	15
Key Zeroization.....	15
Algorithms.....	15
Cryptographic Keys and Critical Security Parameters.....	18
Alternating Bypass Feature.....	21
Key Archiving.....	22
Mitigation of Other Attacks.....	22
FIPS 140-2 Compliant Operation	24
Enabling FIPS-CC mode.....	25
Self-Tests	26
Startup and Initialization Self-tests.....	26
Conditional Self-tests.....	26
Critical Function Self-tests.....	27
Error State.....	27

Overview

This document is a FIPS 140-2 Security Policy for Fortinet Incorporated's FortiGate-VM virtual appliances running FortiOS 5.4. This policy describes how the appliances (hereafter referred to as the 'module') meet 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 1 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 <https://www.fortinet.com/products>.
- Find on-line product support for registered products in the technical support section of the Fortinet corporate website at <https://www.fortinet.com/support>.
- Find contact information for technical or sales related questions in the contacts section of the Fortinet corporate website at <https://www.fortinet.com/contact>.
- Find security information and bulletins in the FortiGuard Center of the Fortinet corporate website at <https://www.fortiguard.com>.

Introduction

FortiGate Virtual Appliances allow you to mitigate blind spots by implementing critical security controls within your virtual infrastructure. They also allow you to rapidly provision security infrastructure whenever and wherever it is needed. FortiGate virtual appliances feature all of the security and networking services common to traditional hardware-based FortiGate appliances

FortiGate virtual appliances detect and eliminate the most damaging, content-based threats from email and Web traffic such as viruses, worms, intrusions, inappropriate Web content and more in real time — without degrading network performance. In addition to providing application level firewall protection, FortiGate virtual appliances deliver a full range of network-level services including VPN, intrusion prevention, web filtering, antivirus, antispam and traffic shaping.

FortiGate virtual appliances support the IPsec industry standard for VPN, allowing VPNs to be configured between a FortiGate virtual appliance and any client or gateway/firewall that supports IPsec VPN. FortiGate virtual appliances also provide SSL VPN services using TLS 1.1 and 1.2.

Security Level Summary

The module meets the overall requirements for a FIPS 140-2 Level 1 validation.

Table 1: Summary of FIPS security requirements and compliance levels

Security Requirement	Compliance Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	3
Roles, Services and Authentication	3
Finite State Model	1
Physical Security	N/A
Operational Environment	1
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	3
Mitigation of Other Attacks	1

Module Descriptions

The FortiGate-VM virtual appliances are software security modules designed to execute on a general purpose computer (GPC) hardware platform. As a software cryptographic module, the virtual appliance has no physical characteristics. The module must rely on physical characteristics of the host system on which it runs. The module supports the physical interfaces of the hardware it is running on. See Figure 1 for a block diagram of the physical system. The module utilizes physical interfaces of the tested platform hosting the virtual environment upon which the module is installed. The hypervisor running on the physical system controls and maps the module's virtual interfaces to the physical interfaces, which include the CPU, memory, network interfaces and hard disk.

----- Cryptographic Boundary

- - - - - Physical Boundary

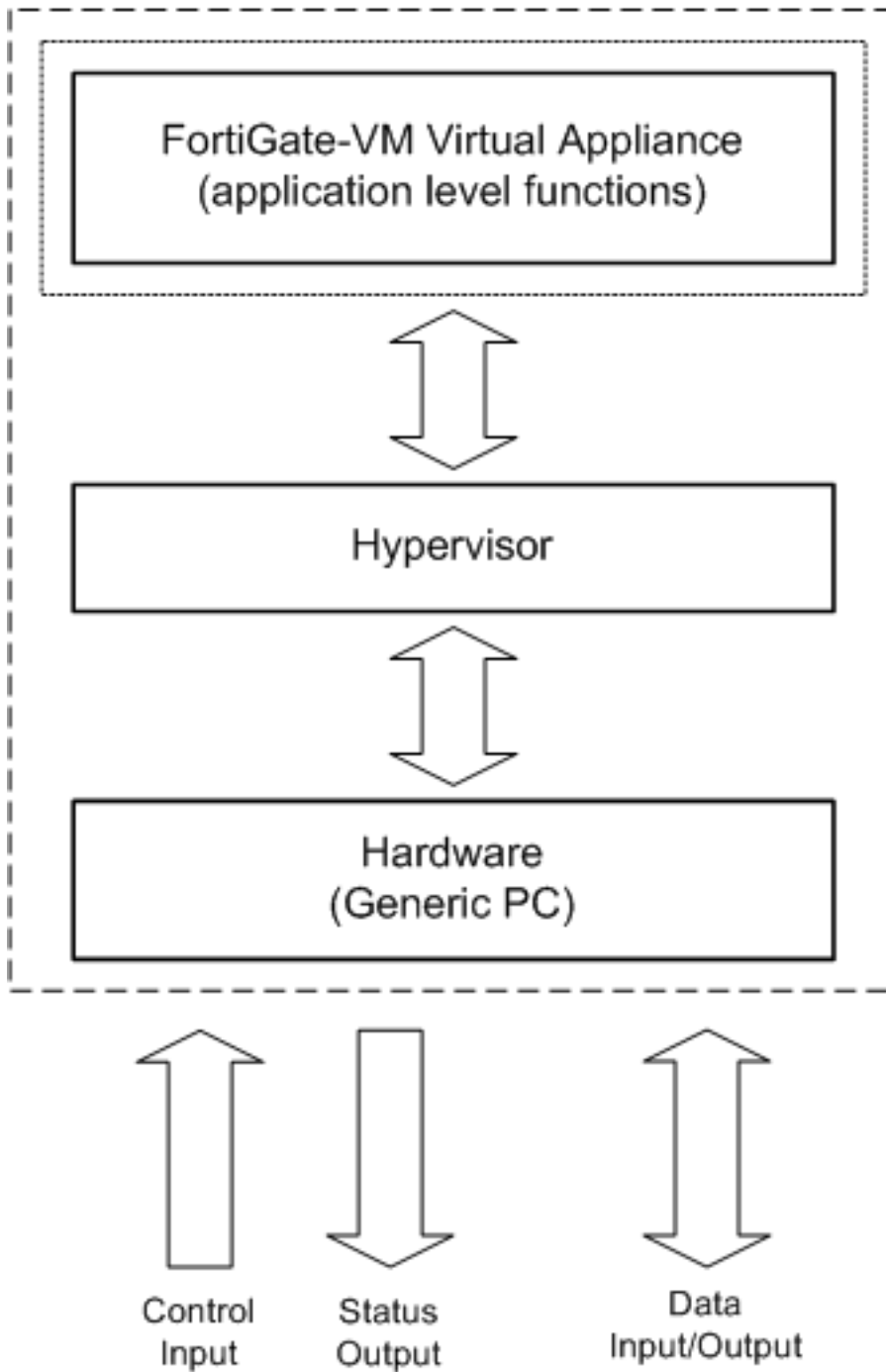


Figure 1: FortiGate-VM Physical and Cryptographic Boundaries

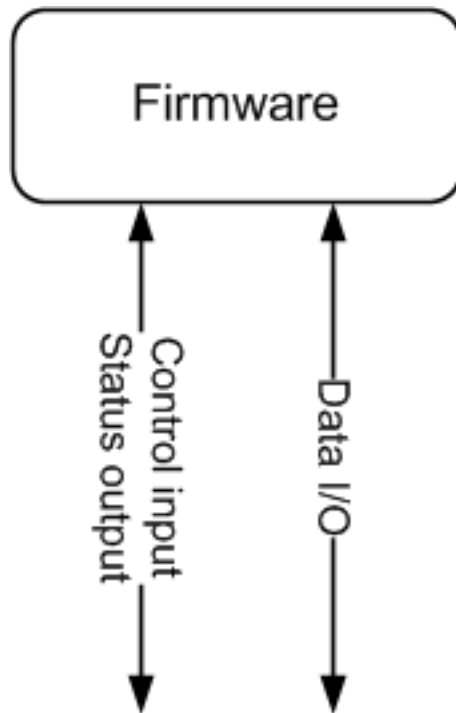


Figure 2: FortiGate-VM logical cryptographic boundary

The validated software version is FortiGate-VM 5.4, b3276, 171006 and a Fortinet entropy token (FTR-ENT-1 or FTR-ENT-2) as the entropy source.

Module Interfaces

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

Table 2: FortiGate-VM logical interfaces and physical ports

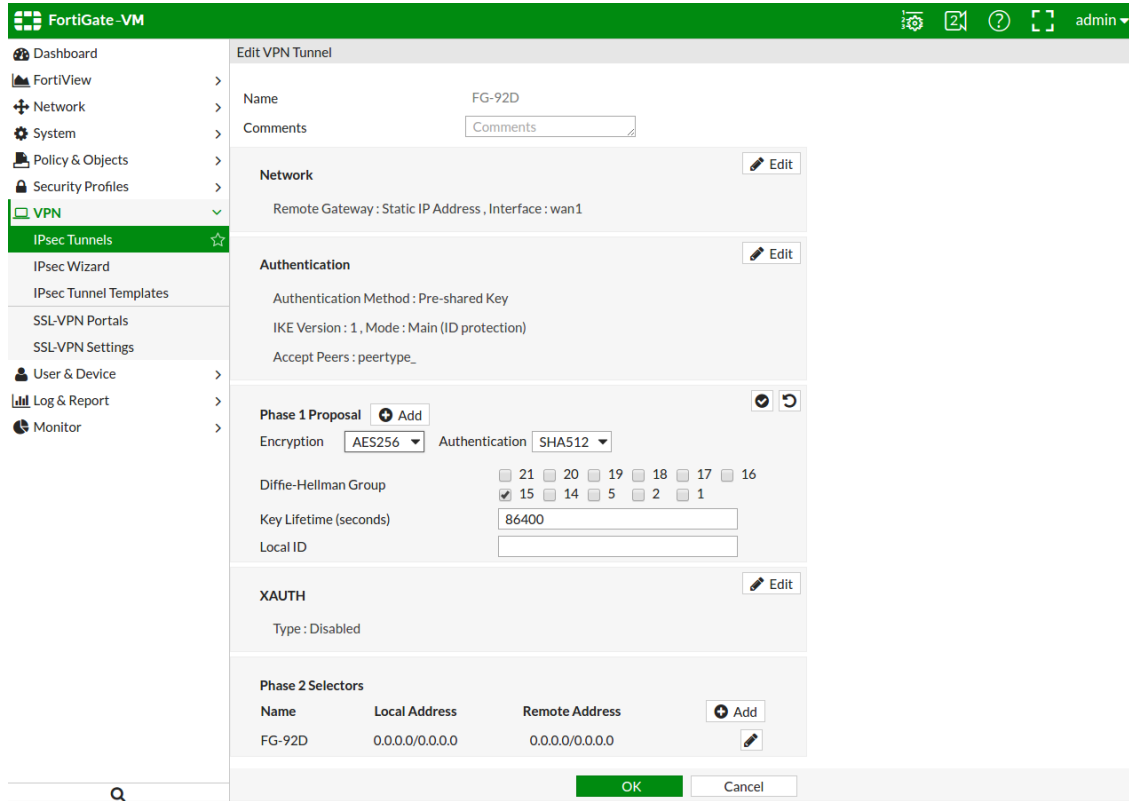
FIPS 140 Interface	FortiGate-VM Interface	Logical Interface	Physical Interface
Data Input	Virtual Ethernet Ports, Virtual USB Ports	API input parameters	Network interface, USB interface (Entropy Token)
Data Output	Virtual Ethernet Ports	API output parameters	Network Interface
Control Input	Virtual Ethernet Ports, Virtual Serial Ports, Virtual USB Ports	API function calls	Network Interface, serial interface, USB interface (USB token)
Status Output	Virtual Ethernet Ports, Virtual Serial Ports	API return values	Network interface, serial interface
Power Input	N/A	N/A	The power supply is the power interface

Web-Based Manager

The FortiGate 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 FortiGate 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 1 - The FortiGate web-based manager



Command Line Interface

The FortiGate 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 FortiHypervisor unit and the management computer. The console connection is a direct serial connection to the FortiHypervisor unit. The serial port on the FortiHypervisor is then mapped to the virtual console connection of the module. 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
- Network 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 modules also provide a **Network User** role for end-users (Users). Network Users can make use of the encrypt/decrypt services, but cannot access the modules for administrative purposes.

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 3: Services available to Crypto Officers

Service	Access	Key/CSP
connect to module locally using the console port	WE	N/A
connect to module remotely using TLS*	WE	Diffie-Hellman Key, EC Diffie Hellman Key, 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

Service	Access	Key/CSP
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	N/A	N/A
show FIPS-CC mode enabled/disabled (console/CLI only)	N/A	N/A
enable FIPS-CC mode of operation (console only)	WE	Configuration Integrity Key
key zeroization	W	All Keys
execute factory reset (disable FIPS-CC mode, console/CLI only)	W	N/A
execute FIPS-CC on-demand self-tests (console only)	E	Configuration Integrity Key, Software Integrity Key
add/delete crypto officers and network users	WE	Crypto Officer Password, Network User Password
set/reset crypto officers and network user passwords	WE	Crypto Officer Password, Network User Password
backup/restore configuration file	RWE	Configuration Encryption Key, Configuration Backup Key
read/set/delete/modify module configuration*	N/A	N/A
execute software update	WE	Software Update Key
read log data	N/A	N/A
delete log data (console/CLI only)	N/A	N/A
execute system diagnostics (console/CLI only)	N/A	N/A
enable/disable alternating bypass mode	N/A	N/A

Service	Access	Key/CSP
read/set/delete/modify IPsec/SSL VPN configuration*	W	IPsec: IPsec Manual Authentication Key, IPsec Manual Encryption Key, IKE Pre-Shared Key, IKE RSA Key, IKE ECDSA Key, Diffie-Hellman Key, EC Diffie-Hellman Key SSL: HTTPS/TLS Server/Host Key, HTTPS/TLS Session Authentication Key, HTTPS/TLS Session Encryption Key
read/set/modify HA configuration	WE	HA Password, HA Encryption Key

Table 4: Services available to Network Users in FIPS-CC mode

Service/CSP	Access	Key/CSP
connect to module remotely using TLS*	WE	Diffie-Hellman Key, EC Diffie-Hellman Key, 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	Network User Password
IPsec VPN controlled by firewall policies*	E	Diffie-Hellman Key, EC Diffie-Hellman Key, all IKE and IPsec Keys, DRBG v and key values, DRBG Output, DRBG Seed, NDRNG Output String
SSL VPN controlled by firewall policies*	E	Network User Password, Diffie-Hellman Key, EC Diffie-Hellman Key, 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

Non-FIPS Approved Services

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

- Configuration backups using password protection
- L2TP and PPTP VPN
- Services marked with an asterisk (*) in Tables 3 and 4 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 implements identity based authentication. Operators must authenticate with a user-id and password combination to access the modules remotely or locally via the console. Remote operator authentication is done over HTTPS (TLS) or SSH. The password entry feedback mechanism does not provide information that could be used to guess or determine the authentication data.

By default, Network User access to the modules is based on firewall policy and authentication by IP address or fully qualified domain names. Network Users can optionally be forced to authenticate to the modules using a username/password combination to enable use of the IPsec VPN encrypt/decrypt or bypass services. For Network Users invoking the SSL-VPN encrypt/decrypt services, the modules support authentication with a user-id/password combination. Network User authentication is done over HTTPS and does not allow access to the modules for administrative purposes.

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 $\{(10) \cdot (26^2) \cdot (32) \cdot (94^4)\}$ which is significantly lower than one in a million.

Note that operator authentication over HTTPS/SSH and Network 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 $\{(10) \cdot (26^2) \cdot (32) \cdot (94^4)\}$ 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 $1/\{(10) \cdot (26^2) \cdot (32) \cdot (94^4)\} / 108,000$ which is less than 1/100,000.

For Network Users invoking the IPsec VPN encrypt/decrypt services, the module acts on behalf of the Network User and negotiates a VPN connection with a remote module. The strength of authentication for IPsec services is based on the authentication method defined in the specific firewall policy: IPsec manual authentication key, IKE pre-shared key, IKE RSA key (RSA certificate) or IKE ECDSA key (ECDSA certificate). The odds of guessing the authentication key for each IPsec method is:

- 1 in 16^{40} for the IPsec Manual Authentication key (based on a 40 digit, hexadecimal key)
- 1 in 94^8 for the IKE Pre-shared Key (based on an 8 character, ASCII printable key)
- 1 in 2^{112} for the IKE RSA Key (based on a 2048bit RSA key size)
- 1 in 2^{128} for the IKE ECDSA Key (based on a P-256 curve ECDSA key size)

Therefore the minimum odds of guessing the authentication key for IPsec is 1 in 94^8 , based on the IKE Pre-shared key.

Operational Environment

The operational environment for the module consists of the FortiOS 5.4 operating system (FortiGate-VM 5.4, b3276, 171006) and the FortiHypervisorOS v1.1.0, build 0129 running on a FortiHypervisor-500D appliance with Intel Xeon E3 processor. No claim can be made as to the correct operation of the

module or the security strengths of the generated keys when ported to an operational environment which is not listed on the validation certificate.

Cryptographic Key Management

Random Number Generation

The modules use a software 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 deleting the FortiGate-VM instance. 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 hypervisor. Executing the following command will output a list of the available internal disks:

```
execute erase-disk ?
```

Algorithms

Table 5: FIPS approved algorithms

Algorithm	NIST Cert Number
CTR DRBG (NIST SP 800-90A) with AES 256-bits	2102

Algorithm	NIST Cert Number
AES in CBC mode (128-, 256-bits)	5404, 5405
AES in GCM mode (128-, 256-bits)	5404, 5405
SHA-1	4338, 4339
SHA-256	4338, 4339
SHA-384	4338, 4339
SHA-512	4338, 4339
HMAC SHA-1	3579, 3580
HMAC SHA-256	3579, 3580
HMAC SHA-384	3579, 3580
HMAC SHA-512	3579, 3580
RSA PKCS1.5 <ul style="list-style-type: none"> • Key Pair Generation: 2048 and 3072-bit • Signature Generation: 2048 and 3072-bit • Signature Verification: 1024, 2048 and 3072-bit • For legacy use, the module supports 1024-bit RSA keys and SHA-1 for signature verification 	2890
ECDSA <ul style="list-style-type: none"> • Key Pair Generation: curves P-256, P-384 and P-521 • Signature Generation: curves P-256, P-384 and P-521 • Signature Verification: curves P-256, P-384 and P-521 	1432 1432 1432, 1433
CVL (SSH) AES 128-bit, AES 256-bit CBC (using SHA1)	1862
CVL (TLS 1.1 and 1.2)	1862
CVL (IKE v1 and v2)	1864
CVL (ECDSA SigGen Component: Curves P-256, P-384 and P-521)	1863, 1865
CKG (NIST SP 800-133)	Vendor Affirmed

KTS (AES Cert. #5404 and HMAC Cert. #3579; 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.

Component functions are used to support protocol implementations. Full cryptographic functions are provided by the supporting algorithms in the cryptographic library.

Table 6: FIPS allowed algorithms

Algorithm
RSA (key wrapping; key establishment methodology provides 112 or 128 bits of encryption strength)
Diffie-Hellman (CVL Certs. #1862 and #1864, key agreement; key establishment methodology provides between 112 and 200 bits of encryption strength)
EC Diffie-Hellman (CVL Certs. #1862 and #1864, key agreement; key establishment methodology provides between 128 and 256 bits of encryption strength)
NDRNG (Entropy Token)
MD5 (used in the TLS protocol only)

Table 7: Non-FIPS approved algorithms

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
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

Note that the IKE, 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 and IKEv2/IPSec.

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.

For IPsec/IKEv2, the GCM implementation meets Option 1 of IG A.5: it is used in a manner compliant with RFCs 4106 and 7296. During operational testing, the module was tested against an independent version of IPsec with IKEv2 and found to behave correctly.

In case the module’s power is lost and then restored, a new key for the AES GCM encryption or decryption shall be established. In addition, when the nonce_explicit part of the IV exhausts the maximum number of values for a session key, this condition triggers a handshake to establish a new encryption key.

Cryptographic Keys and Critical Security Parameters

The following table lists all of the cryptographic keys and critical security parameters used by the modules. The following definitions apply to the table. Note that "Automatic" generation is defined as Electronic Entry/Electronic Distribution as per IG 7.7.

Table 8: Cryptographic Keys and Critical Security Parameters used in FIPS-CC mode

Key or CSP	Generation	Storage	Usage	Zeroization
NDRNG output string	NDRNG	Boot device Plain-text	Input string for the entropy pool (5120-bits)	By erasing the Boot device and power cycling the module
DRBG seed	Internally generated	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	Internally generated	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	Internally generated	Boot device Plain-text	Internal state values for the DRBG	By erasing the Boot device and power cycling the module
IPsec Manual Authentication Key	Manual	Boot device AES encrypted	Used as IPsec Session Authentication Key	By erasing the Boot device and power cycling the module
IPsec Manual Encryption Key	Manual	SDRAM Plain-text	Used as IPsec Session Encryption Key using AES (128-, 256-bit)	By erasing the Boot device and power cycling the module
IPsec Session Authentication Key	Internally generated using DRBG	SDRAM Plain-text	IPsec peer-to-peer authentication using HMAC SHA-1 or HMAC SHA-256	By erasing the Boot device and power cycling the module
IPsec Session Encryption Key	Derived via KDF defined in SP800-135 (IKEv1/IKEv2)	SDRAM Plain-text	VPN traffic encryption/decryption using AES (128-, 256-bit)	By erasing the Boot device and power cycling the module
IKE SKEYSEED	Internally generated using DRBG	SDRAM Plain-text	Used to generate IKE protocol keys	By erasing the Boot device and power cycling the module
IKE Pre-Shared Key	Manual	Boot device AES encrypted	Used to generate IKE protocol keys	By erasing the Boot device and power cycling the module

Key or CSP	Generation	Storage	Usage	Zeroization
IKE Authentication Key	Internally generated using DRBG	SDRAM Plain-text	IKE peer-to-peer authentication using HMAC SHA-1 , -256, -384 or -512	By erasing the boot device and power cycling the module
IKE SA keying material Key	Internally generated using DRBG	SDRAM Plain-text	IPsec SA keying material	By erasing the boot device and power cycling the module
IKE Session Encryption Key	Internally generated via DH or ECDH KAS	SDRAM Plain-text	Encryption of IKE peer-to-peer key negotiation using or AES (128-, 256-bit)	By erasing the boot device and power cycling the module
IKE RSA Key	Externally generated	Boot device Plain-text	RSA private key used in the IKE protocol - used to generate IKE protocol keys (2048- and 3072-bit signatures)	By erasing the boot device and power cycling the module
IKE ECDSA Key	Externally generated	Boot device Plain-text	ECDSA private key used in the IKE protocol - used to generate IKE protocol keys (signatures using P-256, -384 and -521 curves)	By erasing the boot device and power cycling the module
Diffie-Hellman Keys	Internally generated using DRBG	SDRAM Plain-text	Key agreement and key establishment (Public key size of 2048- to 8192-bits with Private key size of 224- to 400-bits)	By erasing the boot device and power cycling the module
EC Diffie-Hellman Keys	Internally generated using DRBG	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
Software Update Key	Preconfigured	Boot device Plain-text	Verification of software integrity when updating to new software versions using RSA public key (software load test, 2048-bit signature)	By erasing the boot device and power cycling the module

Key or CSP	Generation	Storage	Usage	Zeroization
Software Integrity Key	Preconfigured	Boot device Plain-text	Verification of software integrity in the software integrity test using RSA public key (software 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 Session Authentication Key	Internally generated using DRBG	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	Internally generated via DH or ECDH KAS	SDRAM Plain-text	AES (128-, 256-bit) key used for HTTPS/TLS session encryption	By erasing the boot device and power cycling the module
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	Internally generated using DRBG	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	Internally generated via DH or ECDH KAS	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

Key or CSP	Generation	Storage	Usage	Zeroization
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	Preconfigured	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
Network User Password	Manual	Boot device SHA-1 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 FortiGate units in an HA cluster	By erasing the boot device and power cycling the unit
HA Encryption Key	Externally generated	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 software (hardcoded) and are not operator modifiable.

Externally generated keys are generated outside the module and loaded by the operator electronically and are not compliant with SP 800-133 unless they were generated by another FIPS validated module.

Alternating Bypass Feature

The primary cryptographic function of the module is as a firewall and VPN device. The module implements two forms of alternating bypass for VPN traffic: policy based (for IPsec and SSL VPN) and interface based (for IPsec VPN only).

Policy Based VPN

Firewall policies with an action of IPsec or SSL-VPN mean that the firewall is functioning as a VPN start/end point for the specified source/destination addresses and will encrypt/decrypt traffic according

to the policy. Firewall policies with an action of allow mean that the firewall is accepting/sending plaintext data for the specified source/destination addresses.

A firewall policy with an action of accept means that the module is operating in a bypass state for that policy. A firewall policy with an action of IPsec or SSL-VPN means that the module is operating in a non-bypass state for that policy.

Interface Based VPN

Interface based VPN is supported for IPsec only. A virtual interface is created and any traffic routed to the virtual interface is encrypted and sent to the VPN peer. Traffic received from the peer is decrypted. Traffic through the virtual interface is controlled using firewall policies. However, unlike policy based VPN, the action is restricted to Accept or Deny and all traffic controlled by the policy is encrypted/decrypted.

When traffic is routed over the non-virtual interface, the module is operating in a bypass state. When traffic is routed over the virtual interface, the module is operating in a non-bypass state.

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 includes a real-time Intrusion Prevention System (IPS) as well as antivirus protection, antispam and content filtering. Use of these capabilities is optional.

The FortiOS IPS has two components: a signature based component for detecting attacks passing through the FortiGate appliance and a local attack detection component that protects the firewall from direct attacks. Functionally, signatures are similar to virus definitions, with each signature designed to detect a particular type of attack. The IPS signatures are updated through the FortiGuard IPS service. The IPS engine can also be updated through the FortiGuard IPS service.

FortiOS antivirus protection removes and optionally quarantines files infected by viruses from web (HTTP), file transfer (FTP), and email (POP3, IMAP, and SMTP) content as it passes through the FortiGate modules. FortiOS antivirus protection also controls the blocking of oversized files and supports blocking by file extension. Virus signatures are updated through the FortiGuard antivirus service. The antivirus engine can also be updated through the FortiGuard antivirus service.

FortiOS antispam protection tags (SMTP, IMAP, POP3) or discards (SMTP only) email messages determined to be spam. Multiple spam detection methods are supported including the FortiGuard managed antispam service.

FortiOS web filtering can be configured to provide web (HTTP) content filtering. FortiOS web filtering uses methods such as banned words, address block/exempt lists, and the FortiGuard managed content service.

Communications between the module and the FortiGuard servers is done securely over TLS using the FIPS approved algorithms and parameters.

Whenever a IPS, antivirus, antispam or filtering event occurs, the modules can record the event in the log and/or send an alert email to an operator.

For complete information refer to the FortiGate Installation Guide for the specific module in question, the FortiGate Administration Guide and the FortiGate IPS Guide.

FIPS 140-2 Compliant Operation

The FortiGate-VM virtual appliance software 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 and md5sum.txt file from the Fortinet Support site at <https://support.fortinet.com/>
2. Use a hashing utility on the downloaded firmware image to compare and verify the output against the result from the md5sum.txt file.
3. Install the FIPS validated firmware image on the hypervisor.
4. Install the entropy token.
5. Enable the FIPS-CC mode of operation as per the "Enabling FIPS-CC Mode" section.

Additional information can be found in the "FIPS 140-2 and Common Criteria Compliant Operation for FortiOS 5.4" that can be found on the Fortinet technical documentation website at <https://docs.fortinet.com>.

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 FortiGate-VM software. You must ensure that:

- The FortiGate-VM software is configured in the FIPS-CC mode of operation.
- The FortiGate-VM server is installed in a secure physical location.
- Physical access to the FortiGate-VM server is restricted to authorized operators.
- The Fortinet entropy token is enabled.
- The Fortinet entropy token remains in the server 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) characters 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.
- IPsec VPN tunnels using AES-GCM should be configured with a key lifetime of 98,000 KB to ensure a rekey after a maximum of 2^{16} encryptions.

The module can be used in either of its two network operation modes: NAT/Route or Transparent. Note that "mode of operation" in this context does not refer or have any impact on the FIPS approved mode of operation. NAT/Route mode applies security features between two or more different networks (for example, between a private network and the Internet) where the module functions like a network router. Transparent mode applies security features at any point in a network where the module functions like a

network bridge. 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 software 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:

- Software integrity test using RSA 2048-bit signatures
- Configuration/VPN bypass test using HMAC SHA-256
- Triple-DES, CBC mode, encrypt known answer test
- Triple-DES, CBC mode, decrypt known answer test
- 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)
- HMAC SHA-512 known answer test
- SHA-512 known answer test (tested as part of HMAC SHA-512 known answer test)
- RSA signature generation known answer test
- RSA signature verification known answer test
- ECDSA signature generation known answer test
- ECDSA 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
- ECDSA pairwise consistency test
- Configuration/VPN bypass test using HMAC SHA-256
- Software 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

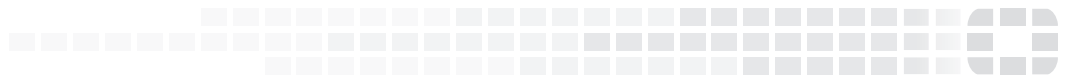
If any of the self-tests or conditional tests fail, the module enters an error state as shown by the console output below:

```
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.



High Performance Network Security



Copyright© 2019 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., in the U.S. and other jurisdictions, and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's General Counsel, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. In no event does Fortinet make any commitment related to future deliverables, features, or development, and circumstances may change such that any forward-looking statements herein are not accurate. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.