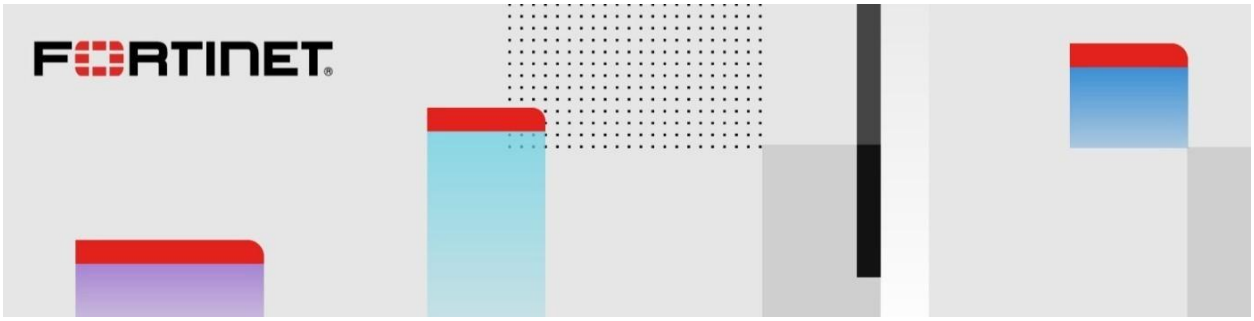
Fortinet, Inc.

FortiGate-VM 7.2/7.4



FIPS 140-3 Non-Proprietary Security Policy

Document Version:	0.2
Publication Date:	August 28, 2026
Software Version:	7.2 (FIPS-CC-72-4) and 7.4 (FIPS-CC-74-2)

FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO GUIDE
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/support-and-training/training.html

NSE INSTITUTE
https://training.fortinet.com

FORTIGUARD CENTER
https://fortiguard.com/

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

FortiGate-VM 7.2/7.4 FIPS 140-3 Level 1 Software Security Policy

01-728-1088100-20260828

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

1 General 6
1.1 Overview 6
1.2 Security Levels 6
1.3 Additional Information 6

2 Cryptographic Module Specification 8
2.1 Description 8
2.2 Tested and Vendor Affirmed Module Version and Identification 9
2.2.1 Executable Code Sets 9
2.2.2 Tested Operational Environments 9
2.2.3 Vendor-Affirmed Operational Environments 10
2.3 Excluded Components 10
2.4 Modes of Operation 10
2.5 Algorithms 11
2.6 Security Function Implementations 13
2.7 Algorithm Specific Information 19
2.8 RBG and Entropy 20
2.9 Key Generation 20
2.10 Key Establishment 20
2.11 Industry Protocols 20
2.12 Additional Information 21

3 Cryptographic Module Interfaces 22
3.1 Ports and Interfaces 22

4 Roles, Services, and Authentication 23
4.1 Authentication Methods 23
4.2 Roles 23
4.3 Approved Services 24
4.4 Non-Approved Services 32
4.5 External Software Loaded 32
4.8 Additional Information 32

5 Software Security 33
5.1 Integrity Techniques 33
5.2 Initiate on Demand 33

6 Operational Environment 34
6.1 Operational Environment Type and Requirements 34
6.2 Configuration Settings and Restrictions 34
6.3 Additional Information 34

7 Physical Security 35

8 Non-Invasive Security 36

9 Sensitive Security Parameters Management 37
9.1 Storage Areas 37
9.2 SSP Input-Output Methods 37
9.3 SSP Zeroization Methods 37
9.4 SSPs 38
9.5 Transitions 45

10 Self-Tests 46
10.1 Pre-Operational Self-Tests 47
10.2 Conditional Self-Tests 47
10.3 Periodic Self-Test Information 49
10.4 Error States 49
10.5 Operator Initiation of Self-Tests 50

11 Life-Cycle Assurance 51

11.1 Installation, Initialization, and Startup Procedures 51
11.2 Administrator Guidance 51
11.3 Non-Administrator Guidance 52
11.4 Design and Rules 52
11.7 Additional Information 52
12 Mitigation of Other Attacks 53
12.1 Attack List 53
12.2 Mitigation Effectiveness 53
12.3 Guidance and Constraints 53

List of Tables

Table 1: Security Levels 6
Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets) 9
Table 3: Tested Operational Environments - Software, Firmware, Hybrid 9
Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid 10
Table 5: Modes List and Description 10
Table 6: Approved Algorithms 13
Table 7: Security Function Implementations 18
Table 8: Entropy Certificates 20
Table 9: Entropy Sources 20
Table 10: Ports and Interfaces 22
Table 11: Authentication Methods 23
Table 12: Roles 23
Table 13: Approved Services 31
Table 14: Storage Areas 37
Table 15: SSP Input-Output Methods 37
Table 16: SSP Zeroization Methods 37
Table 17: SSP Table 1 41
Table 18: SSP Table 2 44
Table 19: Pre-Operational Self-Tests 47
Table 20: Conditional Self-Tests 48
Table 21: Pre-Operational Periodic Information 49
Table 22: Conditional Periodic Information 49
Table 23: Error States 49

List of Figures

Figure 1: Block Diagram 9

1 General

1.1 Overview

This document is a FIPS 140-3 Security Policy for the FortiGate-VM Next Generation Firewall virtual appliances versions 7.2 and 7.4. This policy describes how the appliances (hereafter referred to as the 'module') meet the FIPS 140-3 security requirements and how to operate the module in a FIPS compliant manner. This policy was created as part of the FIPS 140-3 Level 1 validation of the module.

The Federal Information Processing Standards Publication 140-3 - Security Requirements for Cryptographic Modules (FIPS 140-3) details the United States Federal Government requirements for cryptographic modules. Detailed information about the FIPS 140-3 standard and validation program is available on the NIST (National Institute of Standards and Technology) website at https://csrc.nist.gov/projects/cryptographic-module-validation-program

1.2 Security Levels

Section	Title	Security Level
1	General	1
2	Cryptographic module specification	1
3	Cryptographic module interfaces	1
4	Roles, services, and authentication	3
5	Software/Firmware security	1
6	Operational environment	1
7	Physical security	N/A
8	Non-invasive security	N/A
9	Sensitive security parameter management	1
10	Self-tests	1
11	Life-cycle assurance	2
12	Mitigation of other attacks	1
	Overall Level	1

Table 1: Security Levels

1.3 Additional Information

This policy deals specifically with operation and implementation of the modules in the technical terms of the FIPS 140-3 standard and the associated validation program. Other Fortinet product manuals, guides and technical notes can be found at the Fortinet technical documentation website at https://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://support.fortinet.com/.

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://wwww.fortiguard.com.

2 Cryptographic Module Specification

2.1 Description

Purpose and Use:

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 1.2 and 1.3.

Module Type: Software

Module Embodiment: Multi-Chip Standalone

The FortiGate-VM virtual appliances are software security modules designed to execute on a general purpose computer (GPC) hardware platform. The module is a multi-chip standalone cryptographic module. 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. The module binary (.out file) is distributed in an FGT_VM64.ovf or fortios.vmdk file for deployment in VMware ESXi.

Cryptographic Boundary:

The extent of the cryptographic boundary for the module is the FortiGate-VM.

Tested Operational Environment's Physical Perimeter (TOEPP):
The module runs in an ESXi Hypervisor environment which runs on a physical generic PC.

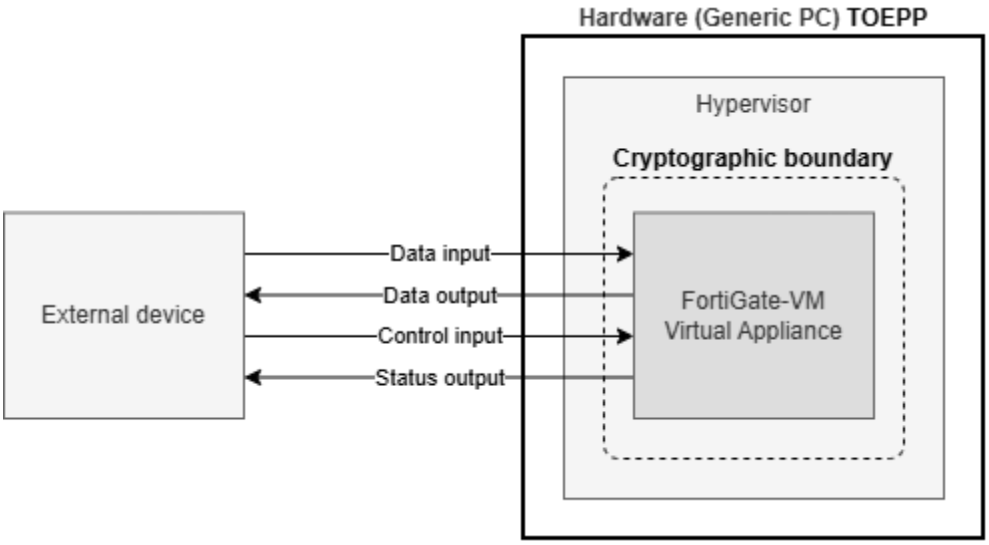


Figure 1: Block Diagram

The validated software version is FIPS-CC-72-4 (7.2) or FIPS-CC-74-2 (7.4). Any software version that is not shown on the module certificate is out of the scope of this validation and requires a separate FIPS 140-3 validation.

Note that 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.

2.2 Tested and Vendor Affirmed Module Version and Identification

2.2.1 Executable Code Sets

Package or File Name	Software/ Firmware Version	Features	Integrity Test
FGT_VM64-v7.2.8FIPS-CC-72-4-build9543-FORTINET.out	FIPS-CC-72-4 (7.2)		Signature verification
FGT_VM64-v7.4.4FIPS-CC-74-2-build8537-FORTINET.out	FIPS-CC-74-2 (7.4)		Signature verification

Table 2: Tested Module Identification – Software, Firmware, Hybrid (Executable Code Sets)

2.2.2 Tested Operational Environments

Operating System	Hardware Platform	Processors	PAA/PAI	Hypervisor or Host OS	Version(s)
FortiOS	PacStar 451	Intel Xeon D-1559	No	VMware ESXi 6.7	FIPS-CC-72-4 (7.2) FIPS-CC-74-2 (7.4)

Table 3: Tested Operational Environments - Software, Firmware, Hybrid

2.2.3 Vendor-Affirmed Operational Environments

Fortinet affirms that the module will operate correctly on Amazon Web Services or Microsoft Azure cloud platforms, provided that those platforms implement one of the vendor-affirmed operational environments listed below:

Operating System	Hardware Platform
Amazon Linux 2027	x86 Architecture (Note: Specific processor/hardware is dependent on Instance/Machine Type selected for operating system)
Amazon Linux 2023	x86 Architecture (Note: Specific processor/hardware is dependent on Instance/Machine Type selected for operating system)
Amazon Linux 2	x86 Architecture (Note: Specific processor/hardware is dependent on Instance/Machine Type selected for operating system)
Azure Linux 3.0	x86 Architecture (Note: Specific processor/hardware is dependent on Instance/Machine Type selected for operating system)
Azure Local v12	x86 Architecture (Note: Specific processor/hardware is dependent on Instance/Machine Type selected for operating system)

Table 4: Vendor-Affirmed Operational Environments - Software, Firmware, Hybrid

2.3 Excluded Components

Each module covered by this security policy comes provisioned with a bootloader tailored to the specific hardware platform. The bootloader does not implement any cryptography, is not used to meet any of the FIPS requirements and is not included in software updates.

The bootloader is excluded from the cryptographic module.

2.4 Modes of Operation

Modes List and Description:

Mode Name	Description	Types of Services	Status Indicator
Approved	Consists of FIPS 140-3 compliant cryptography.	Approved	Implicitly indicated by the successful completion of an approved service

Table 5: Modes List and Description

Mode Change Instructions and Status:

The module only supports approved mode of operation. **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 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.

2.5 Algorithms

Approved Algorithms:

Algorithm	CAVP Cert	Properties	Reference
AES-CBC	A6800, A6831, A6832, A6833	Direction - Decrypt, Encrypt Key Length - 128, 192, 256	SP 800-38A
AES-GCM	A6800, A6831, A6832, A6833	Direction - Decrypt, Encrypt IV Generation - External IV Generation Mode - 8.2.1 Key Length - 128, 256 Tag Length - 128 IV Length - IV Length: 96 Payload Length - Payload Length: 16, 128, 136, 256, 264 AAD Length - AAD Length: 0, 128, 136, 256	SP 800-38D
Counter DRBG	A6800, A6831	Prediction Resistance - No Supports Reseed - Yes Mode - AES-256 Derivation Function Enabled - Yes Additional Input - Additional Input: 0-256 Increment 256 Entropy Input - Entropy Input: 256 Nonce - Nonce: 128 Personalization String Length - Personalization String Length: 0-256 Increment 256 Returned Bits - 256	SP 800-90A Rev. 1
ECDSA KeyGen (FIPS186-5)	A6832, A6833	Curve - P-256, P-384, P-521 Secret Generation Mode - extra bits, testing candidates	FIPS 186-5
ECDSA SigGen (FIPS186-5)	A6832, A6833	Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512 Component - No	FIPS 186-5
ECDSA SigVer (FIPS186-5)	A6832, A6833	Component - No Curve - P-256, P-384, P-521 Hash Algorithm - SHA2-256, SHA2-384, SHA2-512	FIPS 186-5
HMAC-SHA-1	A6832, A6833	MAC - MAC: 32-160 Increment 8 Key Length - Key Length: 8-1024 Increment 8	FIPS 198-1
HMAC-SHA2-224	A6832, A6833	MAC - MAC: 32-224 Increment 8 Key Length - Key Length: 8-1024 Increment 8	FIPS 198-1
HMAC-SHA2-256	A6832, A6833	MAC - MAC: 32-256 Increment 8 Key Length - Key Length: 8-1024 Increment 8	FIPS 198-1

Algorithm	CAVP Cert	Properties	Reference
HMAC-SHA2-384	A6832, A6833	MAC - MAC: 32-384 Increment 8 Key Length - Key Length: 8-1024 Increment 8	FIPS 198-1
HMAC-SHA2-512	A6832, A6833	MAC - MAC: 32-512 Increment 8 Key Length - Key Length: 8-1024 Increment 8	FIPS 198-1
KAS-ECC-SSC Sp800-56Ar3	A6832, A6833	Domain Parameter Generation Methods - P-256, P-384, P-521 Scheme - ephemeralUnified - KAS Role - initiator, responder	SP 800-56A Rev. 3
KAS-FFC-SSC Sp800-56Ar3	A6832, A6833	Domain Parameter Generation Methods - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192 Scheme - dhEphem - KAS Role - initiator, responder	SP 800-56A Rev. 3
KDF IKEv1 (CVL)	A6832, A6833	Authentication Method - Digital Signature, Pre-shared Key, Public Key Encryption Initiator Nonce Length - Initiator Nonce Length: 64-2048 Increment 8 Responder Nonce Length - Responder Nonce Length: 64-2048 Increment 8 Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 224-8192 Increment 8 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512 Preshared Key Length - Preshared Key Length: 8-8192 Increment 8	SP 800-135 Rev. 1
KDF IKEv2 (CVL)	A6832, A6833	Initiator Nonce Length - Initiator Nonce Length: 64-128 Increment 8 Responder Nonce Length - Responder Nonce Length: 64-128 Increment 8 Diffie-Hellman Shared Secret Length - Diffie-Hellman Shared Secret Length: 224-8192 Increment 8 Derived Keying Material Length - Derived Keying Material Length: 160-16384 Increment 8 Hash Algorithm - SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	SP 800-135 Rev. 1
KDF SSH (CVL)	A6832, A6833	Cipher - AES-128, AES-192, AES-256 Hash Algorithm - SHA-1, SHA2-256	SP 800-135 Rev. 1
RSA KeyGen (FIPS186-5)	A6832, A6833	Key Generation Mode - probable Modulo - 2048, 3072, 4096 p mod 8 - 0 Primality Tests - 2powSecStr q mod 8 - 0 Info Generated By Server - No Private Key Format - standard Public Exponent Mode - random	FIPS 186-5

Algorithm	CAVP Cert	Properties	Reference
RSA SigGen (FIPS186-5)	A6832, A6833	Hash Pair - Hash Algorithm - SHA2-256 Salt Length - 32 Mask Function - MGF1 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss	FIPS 186-5
RSA SigVer (FIPS186-5)	A6832	Hash Pair - Hash Algorithm - SHA2-224 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss Public Exponent Mode - random Mask Function - MGF1	FIPS 186-5
RSA SigVer (FIPS186-5)	A6833	Hash Pair - Hash Algorithm - SHA2-256 Salt Length - 32 Mask Function - MGF1 Modulo - 2048, 3072, 4096 Signature Type - pkcs1v1.5, pss Public Exponent Mode - random	FIPS 186-5
Safe Primes Key Generation	A6832, A6833	Safe Prime Groups - ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, MODP-2048, MODP-3072, MODP-4096, MODP-6144, MODP-8192	SP 800-56A Rev. 3
SHA-1, SHA2-224, SHA2-256, SHA2-384, SHA2-512	A6832, A6833	Message Length - Message Length: 0-65528 Increment 8	FIPS 180-4
SHA3-256	A4977	Message Length - Message Length: 0-65536 Increment 8	FIPS 202
TLS v1.2 KDF RFC7627 (CVL)	A6832, A6833	Hash Algorithm - SHA2-256, SHA2-384, SHA2-512 Key Block Length - Key Block Length: 1024	SP 800-135 Rev. 1
TLS v1.3 KDF (CVL)	A6832, A6833	HMAC Algorithm - SHA2-256, SHA2-384 KDF Running Modes - DHE, PSK, PSK-DHE	SP 800-135 Rev. 1

Table 6: Approved Algorithms

2.6 Security Function Implementations

Name	Type	Description	Properties	Algorithms
Configuration backup	BC-Auth MAC	Encryption/decryption and authentication of the configuration backup file		AES-CBC: (A6832, A6833) HMAC-SHA2-256: (A6832)

Name	Type	Description	Properties	Algorithms
				HMAC-SHA2-224: (A6833) SHA2-256: (A6832, A6833)
Configuration integrity	MAC	Calculates MAC of a configuration file		HMAC-SHA2-256: (A6832, A6833) SHA2-256: (A6832, A6833)
ECDSA CSR	AsymKeyPair-KeyGen DigSig-SigGen	Generate a certificate signing request using ECDSA		ECDSA KeyGen (FIPS186-5): (A6832, A6833) ECDSA SigGen (FIPS186-5): (A6832, A6833) SHA2-256: (A6832, A6833)
Entropy	ENT-Cond ENT-ESV	Generate entropy		SHA3-256: (A4977)
KAS-IPSec	DigSig-SigVer KAS-Full	Establishes an IPSec connection with a remote device	IG D.F:Scenario 2, path (2) Key derivation:IG 2.4.B SP 800-135rev1 CVL Key confirmation:No Caveat:Provides between 112 and 256 bits of security strength	ECDSA KeyGen (FIPS186-5): (A6832, A6833) ECDSA SigVer (FIPS186-5): (A6832, A6833) RSA SigVer (FIPS186-5): (A6832, A6833) Safe Primes Key Generation: (A6832, A6833) KAS-ECC-SSC Sp800-56Ar3: (A6832, A6833) KAS-FFC-SSC Sp800-56Ar3: (A6832, A6833) KDF IKEv1: (A6832, A6833) KDF IKEv2: (A6832, A6833) HMAC-SHA2-224: (A6832, A6833) HMAC-SHA2-256:

Name	Type	Description	Properties	Algorithms
				(A6832, A6833) HMAC-SHA2-384: (A6832, A6833) HMAC-SHA2-512: (A6832, A6833) SHA2-224: (A6832, A6833) SHA2-256: (A6832, A6833) SHA2-384: (A6832, A6833) SHA2-512: (A6832, A6833)
KAS-SSH	DigSig-SigVer KAS-Full	Establishes an SSH connection with a remote device	IG D.F:Scenario 2, path (2) Key derivation:IG 2.4.B SP 800-135rev1 CVL Key confirmation:No Caveat:Provides between 112 and 256 bits of security strength	ECDSA KeyGen (FIPS186-5): (A6832, A6833) Safe Primes Key Generation: (A6832, A6833) KAS-ECC-SSC Sp800-56Ar3: (A6832, A6833) KAS-FFC-SSC Sp800-56Ar3: (A6832, A6833) KDF SSH: (A6832, A6833) SHA-1: (A6832, A6833) SHA2-256: (A6832, A6833)
KAS-TLS12	DigSig-SigVer KAS-Full	Establishes a TLS 1.2 connection with a remote device	IG D.F:Scenario 2, path (2) Key derivation:IG 2.4.B SP 800-135rev1 CVL Key confirmation:No Caveat:Provides between 112 and 256 bits of security strength	ECDSA KeyGen (FIPS186-5): (A6832, A6833) ECDSA SigVer (FIPS186-5): (A6832, A6833) RSA SigVer (FIPS186-5): (A6832, A6833) Safe Primes Key Generation: (A6832, A6833) KAS-ECC-SSC Sp800-56Ar3: (A6832, A6833)

Name	Type	Description	Properties	Algorithms
				KAS-FFC-SSC Sp800-56Ar3: (A6832, A6833) TLS v1.2 KDF RFC7627: (A6832, A6833) HMAC-SHA-1: (A6832, A6833) HMAC-SHA2-256: (A6832, A6833) HMAC-SHA2-384: (A6832, A6833) SHA-1: (A6832, A6833) SHA2-256: (A6832, A6833) SHA2-384: (A6832, A6833)
KAS-TLS13	DigSig-SigVer KAS-Full	Establishes a TLS 1.3 connection with a remote device	IG D.F:Scenario 2, path (2) Key derivation:IG 2.4.B SP 800-135rev1 CVL Key confirmation:No Caveat:Provides between 112 and 256 bits of security strength	ECDSA KeyGen (FIPS186-5): (A6832, A6833) ECDSA SigVer (FIPS186-5): (A6832, A6833) RSA SigVer (FIPS186-5): (A6832, A6833) Safe Primes Key Generation: (A6832, A6833) KAS-ECC-SSC Sp800-56Ar3: (A6832, A6833) KAS-FFC-SSC Sp800-56Ar3: (A6832, A6833) TLS v1.3 KDF: (A6832, A6833) HMAC-SHA2-256: (A6832, A6833) HMAC-SHA2-384: (A6832, A6833) SHA2-256: (A6832, A6833) SHA2-384: (A6832, A6833)

Name	Type	Description	Properties	Algorithms
Key decrypt	BC-UnAuthDecrypt	Decrypting of static keys used for authentication in network protocols		AES-CBC: (A6832, A6833)
Key encrypt	BC-UnAuthEncrypt	Encryption of the pre-shared key used for reestablishing IPSec tunnels.		AES-CBC: (A6832, A6833)
Operator authentication	SHA	Generates a hash of the operator password		SHA2-256: (A6832, A6833)
RSA CSR	AsymKeyPair-KeyGen DigSig-SigGen	Generate a certificate signing request using RSA		RSA KeyGen (FIPS186-5): (A6832, A6833) RSA SigGen (FIPS186-5): (A6832, A6833) SHA2-256: (A6832, A6833)
Random number	DRBG	Generate random number using the DRBG		Counter DRBG: (A6800, A6831) AES-CBC: (A6800, A6831)
SW Authentication	DigSig-SigVer	Verifies the signature on software		RSA SigVer (FIPS186-5): (A6832, A6833) SHA2-256: (A6832, A6833)
Transfer data with CBC	BC-AuthDecrypt BC-AuthEncrypt	Encrypt and decrypt network packets over the IPSec, SSH or TLS protocols when a non-AEAD cipher suite is negotiated.		AES-CBC: (A6832, A6833) HMAC-SHA-1: (A6832, A6833) HMAC-SHA2-256: (A6832, A6833) HMAC-SHA2-384: (A6832, A6833) HMAC-SHA2-512: (A6832, A6833) SHA-1: (A6832, A6833) SHA2-256: (A6832, A6833) SHA2-384: (A6832, A6833) SHA2-512: (A6832, A6833)

Name	Type	Description	Properties	Algorithms
Transfer data with GCM	BC-AuthDecrypt BC-AuthEncrypt	Encrypt and decrypt network packets over the IPSec, SSH or TLS protocols when an AEAD cipher suite is negotiated.		AES-GCM: (A6800, A6831, A6832, A6833)
CKG	CKG	Generation of seeds for asymmetric key generation		CKG: ()

Table 7: Security Function Implementations

2.7 Algorithm Specific Information

AES-GCM

FIPS140-3 IG C.H, Scenario 1

TLS 1.2: The module supports TLS 1.2 GCM Cipher Suites listed in SP800-52 Rev2, Section 3.3.1 and defined in RFCs 5246, 5288, 5289, 8422.

During FIPS testing, the module's implementation of TLS 1.2 was successfully tested against an independently developed instance of TLS 1.2. This demonstrated that the module's implementation is compliant with the following aspects of RFC 5246:

- the counter portion of the IV is set by the Module within its cryptographic boundary.
- when the nonce_explicit portion of the IV reaches the maximum possible value (2^{32}-1), a rekeying is triggered

The IV is only used in the context of TLS 1.2.

TLS 1.3: The module supports TLS 1.3 GCM Cipher Suites listed in SP800-52 Rev2, Section 3.3.1 and defined in RFC 8446. During FIPS testing, the module's implementation of TLS 1.3 was successfully tested against an independently developed instance of TLS 1.3. This demonstrated that the module's implementation is compliant with the following aspects of RFC 8446:

- the client_write_iv and server_write_iv values are derived entirely within the module from the application_traffic_secret,
- when the sequence_number reaches the maximum possible value (2^{64}-1), a rekeying is triggered

The IV is only used in the context of TLS 1.3.

The module does not persistently store SSPs specific to a TLS session. In the event that the module’s power is lost and then restored, new keys and IVs for use with AES-GCM within TLS are established.

IPsec-v3: If IKEv2 is used, the Module is compliant with RFC 4106 and/or RFC 5282 (depending on the protocols supporting GCM). The module uses an RFC 7296 compliant KDF to establish the shared secret SKEYSEED from which the AES-GCM encryption keys are derived.

When:

- the module exhausts the maximum number of possible values for a given security association, or
- the IV exhausts the maximum number of possible values (e.g., a 64-bit counter starting from 0 and increasing, when it reaches the maximum value of 2^{64}-1)

the module will trigger a rekeying with IKEv2 to establish a new encryption key for the security association.

In case the module’s power is lost and then restored, a new key for use with the AES-GCM encryption/decryption is established.

SSHv2: This module is compliant with RFCs 4252, 4253 and the rules for using AES-GCM documented in RFC 5647. The IV is only used in the context of the AES-GCM mode encryptions within the SSHv2 protocol.

No more than 2^{64}–1 AES encryptions may be performed in the same session. The session is automatically terminated once this number of encryptions is reached. If the invocation counter reaches its maximum value 2^{64}–1, the next AES-GCM encryption is performed with the invocation counter set to 0.

When a session is terminated for any reason, the keys and IVs are zeroised and cannot be used again.

In case the module's power is lost and then restored, new keys and IVs for use with the AES-GCM are established

2.8 RBG and Entropy

Cert Number	Vendor Name
E139	Fortinet

Table 8: Entropy Certificates

Name	Type	Operational Environment	Sample Size	Entropy per Sample	Conditioning Component
FortiOS CPU Jitter Entropy library 1.0	Non-Physical	Intel Xeon D-1559	256	256	SHA3-256 (A4977)

Table 9: Entropy Sources

The module uses FortiOS CPU Jitter Entropy Library 1.0 to seed the DRBG during the modules' boot process and to periodically reseed the DRBG. The entropy loaded into the approved AES-256 bit DRBG is 256 bits. The entropy source is over-sampled and then an SHA3-256 post-conditioning component is applied. The reseed interval (number of requests between reseeds) of CTR_DRBG in FortiGate Linux kernel is 100000 (within the maximum reseed interval of 2^{48} specified for this CTR_DRBG as per SP800-90Arev1).

The module uses CTR_DRBG as per section 10.2.1 of SP 800-90Arev1 to generate random numbers.

2.9 Key Generation

The module implements asymmetric key generation services compliant with FIPS 186-5 to generate RSA and ECDSA keys. Seeds for these services are generated from the unmodified output of the module's approved DRBG.

2.10 Key Establishment

The module implements the following approved key agreement methods (per 140-3 IG D.F):

Elliptic curves

The module establishes EC DH shared secrets compliant to SP 800-56Arev3, using elliptic-curves specified in Appendix D of SP 800-186.

Finite fields

The module establishes DH shared secrets compliant to SP 800-56Arev3, using safe prime groups listed in Appendix D of SP 800-56Arev3.

2.11 Industry Protocols

The module implements compliant key derivation functions (KDFs) as part of its implementation of the following protocols:

- TLS 1.2
- TLS 1.3
- IKEv1

- IKEv2
- SSH
- SNMP

No parts of these protocols, other than the approved cryptographic algorithms and the KDFs, have been tested by the CAVP and CMVP.

2.12 Additional Information

When the module is powered on, it initiates self-tests against the algorithms claimed in section 2.5. Once all the tests are passed, the module continues to function in the approved mode. Please refer to section 11.1 start up procedure for installation process.

3 Cryptographic Module Interfaces

3.1 Ports and Interfaces

Physical Port	Logical Interface(s)	Data That Passes
N/A	Data Input	Network traffic, passwords, user data, new software images, configuration data, configuration file
N/A	Data Output	Network traffic, configuration file
N/A	Control Input	Commands
N/A	Status Output	Module status and configuration information

Table 10: Ports and Interfaces

FortiGate's logical interfaces and physical interfaces are described above.

The module does not implement a control output interface.

4 Roles, Services, and Authentication

4.1 Authentication Methods

Method Name	Description	Security Mechanism	Strength Each Attempt	Strength per Minute
IPSec	Establishing a secure connection over IPSec	KAS-IPSec	1 in 5.2x10^33	1 in 6.6x10^24
Password	Username and password	Operator authentication	1 in 2.8x10^15	1 in 111262

Table 11: Authentication Methods

The module implements identity-based authentication. To access management services the module enforces SFA (Single factor authentication) mechanism (i.e. username and password). The minimum password length is 8 characters (maximum password length is 128 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.

When connecting to the module over IPSec, the module authenticates the other device using a public key certificate. The other device must minimally provide a 2048-bit RSA public key, providing 112-bits of security strength.

For both authentication methods, the strength-per-minute is computed based on the maximum throughput of the module's 25 Gigabit network ports.

4.2 Roles

Name	Type	Operator Type	Authentication Methods
Admin	Identity	CO	Password
User	Identity	User	None

Table 12: Roles

The module provides the following roles:

Crypto Officer (CO)

The 'Admin' operator is initially assigned to the Crypto-officer role. A Crypto Officer has read-write- execute access to all the module’s administrative services. The initial Crypto Officer can create accounts for additional operators. These additional operators are also assigned the Crypto Officer role and can be assigned a range of read-write-execute or read only access permissions including the ability to create accounts.

Network User

The module also supports a Network User (or just 'User') role. All operators that establish a connection to the module are assigned the Network User role. Network Users can make use of the network traffic services, but cannot access the module for administrative purposes.

The module does not provide a Maintenance role.

4.3 Approved Services

The following tables detail the types of 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:

- G – Generate - The module generates or derives the SSP.
- R – Read - The SSP is read from the module (e.g. the SSP is output).
- W – Write - The SSP is updated, imported, or written to the module.
- E – Execute - The module uses the SSP in performing a cryptographic operation.
- Z – Zeroise - The module zeroises the SSP

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Add/delete operator	Add or delete operator account	System Logs	User data	Status	None	Admin - Crypto officer password: Z
Authenticate to Module	Crypto Officer authenticates to the module over CLI/GUI	System Logs	Password	Status	Operator authentication	User - Crypto officer password: W
Backup/restore configuration file	Backup/restore configuration file	System Logs	Command	Status	Configuration backup	Admin - Configuration encryption key: E - Configuration backup key: E - Crypto officer password: None
Connect the module to remote FortiAnalyzer for transferring module's local logs to FortiAnalyzer using TLS	Connect to FortiAnalyzer for log transfer via OFTP over TLS	System Logs	Command	Status	KAS-TLS12 Random number CKG	Admin - DH Private Key: G,E,Z - DH Public Key: G,R,E,Z - ECDH Private Key : G,E,Z - ECDH Public Key : G,R,E,Z - TLS-PMS: G,E,Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- TLS12-MS: G,E,Z - TLS-SAK: G - TLS-SEK: G - DRBG Output: E - DH Peer Public Key: W,E,Z - ECDH Peer Public Key : W,E,Z - TLS Client key: W,E,Z - DH-SS: G - ECDH-SS: G - GCM-IV: G
Connect to module remotely for web admin using SSH	The module offers authentication over console port	System Logs	Command	Status	Entropy KAS-SSH Random number CKG	Admin - DH Private Key: G,E,Z - DH Public Key: G,R,E,Z - ECDH Private Key : G,E,Z - ECDH Public Key : G,R,E,Z - SSH-SS: G,E,Z - SSH-IK: G - SSH-EK: G - DRBG Output: E - DH Peer Public Key: W,E,Z - ECDH Peer Public Key : W,E,Z - SSH client public key: W,E,Z - SSH server public key: R - DH-SS: G - ECDH-SS: G
Connect to module remotely for web admin using TLS 1.2	The module offers authentication over TLS 1.2	System Logs	Command	Status	KAS-TLS12 Random number CKG	User - DH Private Key: G,E,Z - DH Public Key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						G,R,E,Z - ECDH Private Key : G,E,Z - ECDH Public Key : G,R,E,Z - TLS13-HS: G,E,Z - TLS13-HTS: G,E,Z - TLS13-MS: G,E,Z - TLS13-TK: G - DRBG Output: E - DH Peer Public Key: W,E,Z - ECDH Peer Public Key : W,E,Z - TLS Client key: W,E,Z - TLS Server Public Key: R - DH-SS: G - ECDH-SS: G - GCM-IV: G
Connect to module remotely for web admin using TLS 1.3	The module offers authentication over TLS 1.3	System Logs	Command	Status	KAS-TLS13 Random number CKG	User - DH Private Key: G,E,Z - DH Public Key: G,R,E,Z - ECDH Private Key : G,E,Z - ECDH Public Key : G,R,E,Z - TLS-PMS: G,E,Z - TLS12-MS: G,E,Z - TLS-SAK: G - TLS-SEK: G - DRBG Output: E - DH Peer Public Key: W,E,Z - ECDH Peer Public Key : W,E,Z - TLS Client key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						W,E,Z - TLS Server Public Key: R
Enable approved mode of operation (Console only)	Enable approved mode of operation	System Logs	Command	Status	Configuration integrity	Admin - Configuration integrity key: E
Enable/disable alternating bypass mode	Configure the IPSec VPN policies	System Logs	Command	Status	Configuration integrity	Admin - Configuration integrity key: E
Establish IPsec connection	Establish IPsec connection	System Logs	Command	Status	KAS-IPSec Random number CKG	User - DH Private Key: G,E,Z - DH Public Key: G,R,E,Z - ECDH Private Key : G,E,Z - ECDH Public Key : G,R,E,Z - IKE-PSK: G,E - SKEYID: G,E,Z - SKEYID_e: G - SKEYID_a: G - DRBG Output: E - DH Peer Public Key: W,E,Z - ECDH Peer Public Key : W,E,Z - IKE RSA Peer Public key: W,E,Z - IKE ECDSA Peer Public Key: W,E,Z - SKEYSEED: G,E,Z - SK_a: G - SK_e: G - IKE RSA Public Key: R - IKE ECDSA Public Key: R

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						- DH-SS: G - ECDH-SS: G
Execute FIPS-CC on-demand self-tests (console only)	Execute self tests manually	System Logs	Command	Status	SW Authentication	Admin - Software update key: E
Execute factory reset (disable approved mode, Console only)	Zeroise the module	System Reboot	Command	-	None	Admin - Crypto officer password: Z - Configuration integrity key: Z - Configuration encryption key: Z - Configuration backup key: Z - Software update key: Z - IPSec Manual Authentication key: Z - IPsec Manual Encryption key: Z - IKE RSA Private Key: Z - IKE RSA Public Key: Z - IKE ECDSA Private Key: Z - IKE ECDSA Public Key: Z - HA Password: Z - HA Encryption key: Z - TLS Server Private Key: Z - TLS Server Public Key: Z - SSH server private key: Z - SSH server public key: Z

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Execute software update	Execute software update	System Logs	SW image	Status	SW Authentication	Admin - Software update key: E
Format log disk (console only)	Format log disk	System Logs	Command	Status	None	Admin
Generate CSR	Generate certificate signing request	System Logs	Command	CSR	ECDSA CSR RSA CSR Random number CKG	Admin - IKE RSA Private Key: G,E - IKE RSA Public Key: G,R,E - IKE ECDSA Private Key: G,E - IKE ECDSA Public Key: G,R,E - DRBG Output: E
Generate entropy	Generate entropy for seeding the DRBG (Cannot be operator invoked)	Implicit	Command	Entropy	Entropy	Unauthenticated - Entropy input: G
Generate random number	Generate random number for other module service (Cannot be operator invoked)	Implicit	Entropy	Random number	Random number	Unauthenticated - Entropy input: E - DRBG Seed: G - DRBG V: G,E - DRBG Key: G,E - DRBG Output: G
Modify operator preferences	Modify operator preferences	System Logs	User data	Status	None	Admin
Read log data	Read log data	System Logs	Command	Log data	None	Admin
Read/set/delete/modify HA configuration	Configure the module's high availability ports	System Logs	Configuration data	Status	Configuration integrity	Admin - HA Password: W - HA Encryption key: W - Configuration integrity key: E
Read/set/delete/modify IPSec/SSL VPN configuration	Configure the module's VPN	System Logs	Configuration data	Status	Configuration integrity	Admin - IPSec Manual Authentication key:

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
						W - IPsec Manual Encryption key: W
Read/set/delete/modify configuration	Configure the module	System Logs	Configuration data	Status	Configuration integrity	Admin
Set/reset operator password	Sets or resets an operator password	System Logs	Command, password	Status	None	Admin - Crypto officer password: W
Show approved mode enabled (console/cli only)	Display the module's mode	System Logs	Command	Status	None	Admin
Show system status	Show system status	Status displayed via CLI/GUI	Command	Status	None	Admin
Show version	Display the version of the module	Status displayed via CLI/GUI	Command	Status	None	Admin
Traffic over HTTP	Traffic over HTTP	Implicit	HTTP request	HTTP data	Transfer data with GCM Transfer data with CBC	User - TLS-SAK: E - TLS-SEK: E - TLS13-TK: E - GCM-IV: E
Traffic over IPsec	Traffic over IPsec	Implicit	Traffic data	Traffic data	Transfer data with GCM Transfer data with CBC	User - SKEYID_a: E - SKEYID_e: E - SK_a: E - SK_e: E - IPSec Manual Authentication key: E - IPsec Manual Encryption key: E - GCM-IV: E
Traffic over SSH	Traffic over SSH	Implicit	SSH Command	Status	Transfer data with GCM Transfer data with CBC	Admin - SSH-IK: E - SSH-EK: E - GCM-IV: E

Name	Description	Indicator	Inputs	Outputs	Security Functions	SSP Access
Key encrypt	Encryption of the pre-shared key used for reestablishing IPSec tunnels. (Cannot be operator invoked)	Implicit	-	-	Key encrypt	User - Configuration encryption key: E - IKE-PSK: None
Key decrypt	Decryption of static keys used for authentication in network protocols. Decryption of previously established IPSec pre shared key. (Cannot be operator invoked)	Implicit	-	-	Key decrypt	Admin - Configuration encryption key: E - IPSec Manual Authentication key: None - IKE-PSK: None - IKE RSA Private Key: None - IKE RSA Public Key: None - IKE ECDSA Private Key: None - IKE ECDSA Public Key: None - HA Encryption key: None - HA Password: None - SSH server private key: None - SSH server public key: None

Table 13: Approved Services

4.4 Non-Approved Services

The module does not provide any non-approved services.

4.5 External Software Loaded

The module supports the updating of the entire software image. Software images are transferred into the host device by the crypto officer from the administrative GUI over HTTPS. On load, the new software image is authenticated using an RSA 2048-bit digital signature verification. A successful verification will trigger an automatic reboot of the module resulting in the zeroisation of all SSPs.

Any software loaded into this module that is not shown on the module certificate, is out of the scope of this validation and requires a separate FIPS 140-3 validation.

4.8 Additional Information

The module supports two methods of handling VPN traffic: policy based (for IPsec and SSL VPN) and interface based (for IPsec VPN only).

Policy Based VPN

Firewall policies with 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 accept mean that the firewall is accepting/sending plaintext data for the specified source/destination addresses.

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.

5 Software Security

5.1 Integrity Techniques

The module uses RSA 2048-bit digital signature as an approved integrity technique for the verification of software. The integrity test runs when the host device starts.

5.2 Initiate on Demand

The integrity test can be invoked on demand from the CLI using:

```
execute fips kat Firmware-integrity
```

6 Operational Environment

6.1 Operational Environment Type and Requirements

The operational environment for the module consists of the FortiGate-VM software and VMware ESXi 6.7 hypervisor running on the appliances listed below. The underlying OS is FortiOS 7.2 or 7.4. 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.

Type of Operational Environment: Modifiable

How Requirements are Satisfied:

ESXi provides for a single separate VM instance on the host such that the environment is isolated and process space separation is maintained. This meets the FIPS 140-3 requirement.

6.2 Configuration Settings and Restrictions

None.

6.3 Additional Information

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) 1.2 or 1.3 is required for remote access to the web-based manager when the module is operating in the approved mode.

7 Physical Security

Not Applicable. FortiGate-VM is a level 1 software module.

8 Non-Invasive Security

The Module does not implement any mitigation methods against non-invasive attacks.

9 Sensitive Security Parameters Management

9.1 Storage Areas

Storage Area Name	Description	Persistence Type
Software image	Embedded in the software image	Static
RAM	SRDAM of host device	Dynamic

Table 14: Storage Areas

9.2 SSP Input-Output Methods

Name	From	To	Format Type	Distribution Type	Entry Type	SFI or Algorithm
Public Key import	External network device	RAM	Plaintext	Automated	Electronic	
Password	External network device	RAM	Plaintext	Manual	Electronic	
Preloaded	Manufacturer	Software image	Plaintext	N/A	N/A	
Public Key export	RAM	External network device	Plaintext	Automated	Electronic	

Table 15: SSP Input-Output Methods

9.3 SSP Zeroization Methods

Zeroization Method	Description	Rationale	Operator Initiation
Auto	Automatic zeroisation of SSPs that are no longer needed.	Memory is overwritten with 0's	N/A
Reboot	Power-cycle the host device	All RAM is zeroised by a power-cycle.	Unplugging the device
Service	Invoking the service to zeroise all SSPs in persistent storage. Causes an automatic reboot of the host device that zeroises all SSPs in volatile memory.	The CLI command overwrites the storage location keys with 0's, making them irretrievable	Issue the CLI command : execute erase-disk

Table 16: SSP Zeroization Methods

Reconfiguring the module such that FIPS-CC Mode is disable, erases the current configuration and zeroizes most keys and critical security parameters. Persistently stored CSPs are zeroized by erasing the module's boot device by issuing the following CLI command:

execute factoryreset

Upon completion of the erase operation, the module automatically triggers a power cycle of the host device. This results in the zeroisation of all SSPs stored in volatile RAM. The rebooting of the module indicates that zeroisation has been successful.

9.4 SSPs

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
Configuration backup key	Used to authenticate backups of system configurations	256 - 128	Authentication - CSP			Configuration backup
Configuration encryption key	Used to encrypt sensitive configuration data	256 - 128	Symmetric - CSP			Configuration backup Key decrypt Key encrypt
Configuration integrity key	Used to ensure the integrity and authenticity of configuration data	256 - 128	Authentication - CSP			Configuration integrity
Crypto officer password	Password used by CO to manage cryptographic operations	min 64 - -	Password - CSP			
DH Peer Public Key	Imported peer public key for DH key exchange	2048-8192 - 112-200	Asymmetric - PSP			KAS-IPSec KAS-SSH KAS-TLS12 KAS-TLS13
DH Private Key	Ephemeral private key for DH key exchange	2048-8192 - 112-200	Asymmetric - CSP	KAS-IPSec KAS-SSH KAS-TLS12 KAS-TLS13		KAS-IPSec KAS-SSH KAS-TLS12 KAS-TLS13
DH Public Key	Ephemeral public key for DH key exchange	2048-8192 - 112-200	Asymmetric - PSP	KAS-IPSec KAS-SSH KAS-TLS12 KAS-TLS13		
DH-SS	Shared secret computed from DH keys. Used as TLS 1.2 pre-master secret, TLS 1.3 master secret, SSH shared secret or IKE shared secret	2048-8192 - 112-200	Asymmetric - CSP		KAS-IPSec KAS-SSH KAS-TLS12 KAS-TLS13	KAS-IPSec KAS-SSH KAS-TLS12 KAS-TLS13
DRBG Key	Internal state value for the DRBG	256 - -	DRBG - CSP	Random number		Random number
DRBG Output	Random numbers used in cryptographic algorithms	256 - 256	DRBG - CSP	Random number		Random number
DRBG Seed	Seed used by the DRBG	384 - 256	DRBG - CSP	Random number		Random number

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
DRBG V	Internal state value for the DRBG	128 - -	DRBG - CSP	Random number		Random number
ECDH Peer Public Key	Imported peer public key for ECDH key exchange	P-256/P-384/P-521 - 128/192/256	Asymmetric - PSP			KAS-IPSec KAS-SSH KAS-TLS12 KAS-TLS13
ECDH Private Key	Ephemeral private key for ECDH key exchange	P-256/P-384/P-521 - 128/192/256	Asymmetric - CSP	KAS-IPSec KAS-SSH KAS-TLS12 KAS-TLS13		KAS-IPSec KAS-SSH KAS-TLS12 KAS-TLS13
ECDH Public Key	Ephemeral public key for ECDH key exchange	P-256/P-384/P-521 - 128/192/256	Asymmetric - PSP	KAS-IPSec KAS-SSH KAS-TLS12 KAS-TLS13		
ECDH-SS	Shared secret computed from ECDH keys. Used as TLS 1.2 pre-master secret, TLS 1.3 master secret, SSH shared secret or IKE shared secret	256-521 - 128-256	Asymmetric - CSP		KAS-IPSec KAS-SSH KAS-TLS12 KAS-TLS13	KAS-IPSec KAS-SSH KAS-TLS12 KAS-TLS13
Entropy input	DRBG Input from the entropy pool	256 - 256	DRBG - CSP	Entropy		Random number
Software update key	Used to verify the new software load	2048 bits - 112 bits	Asymmetric - PSP			SW Authentication
HA Encryption key	Encryption of traffic between units in an HA cluster	128 - 128	Symmetric - CSP			
HA Password	Used to authenticate FortiGate units in an HA cluster	min 64 - -	Password - CSP			
IKE ECDSA Peer Public Key	Peer authentication key	P-256/P-384/P-521 - 128/192/256	Asymmetric - PSP			KAS-IPSec
IKE ECDSA Private Key	Module authentication key	P-256/P-384/P-521 - 128/192/256	Asymmetric - CSP			KAS-IPSec
IKE ECDSA Public Key	Module authentication key	P-256/P-384/P-521 - 128/192/256	Asymmetric - PSP			KAS-IPSec
IKE RSA Peer Public key	Peer authentication key	2048, 3072 - 112, 128	Asymmetric - PSP			KAS-IPSec

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
IKE RSA Private Key	Module authentication key	2048, 3072 - 112, 128	Asymmetric - CSP			KAS-IPSec
IKE RSA Public Key	Module authentication key	2048, 3072 - 112, 128	Asymmetric - PSP			KAS-IPSec
IKE-PSK	IKE Pre-shared key	>112 - >112	Symmetric - CSP	KAS-IPSec		KAS-IPSec
IPSec Manual Authentication key	Used as IPsec Session Authentication Key	160 to 512 - 112 to 256	Symmetric - CSP			Transfer data with CBC
IPsec Manual Encryption key	Used as IPsec Session Encryption Key	160 to 512 - 112 to 256	Symmetric - CSP			Transfer data with GCM Transfer data with CBC
SKEYID	IKEv1 key derivation key	160/224/256/384/512 - 160/224/256	Symmetric - CSP	KAS-IPSec		KAS-IPSec
SKEYID_a	IKEv1 session authentication key	160 to 512 - 112 to 256	Symmetric - CSP	KAS-IPSec		Transfer data with CBC
SKEYID_e	IKEv1 session encryption key	160 to 512 - 112 to 256	Symmetric - CSP	KAS-IPSec		Transfer data with GCM Transfer data with CBC
SKEYSEED	IKEv2 key derivation key	160/224/256/384/512 - 160/224/256	Symmetric - CSP	KAS-IPSec		KAS-IPSec
SK_a	IKEv2 session authentication key	160 to 512 - 112 to 256	Symmetric - CSP	KAS-IPSec		Transfer data with CBC
SK_e	IKEv2 session encryption key	160 to 512 - 112 to 256	Symmetric - CSP	KAS-IPSec		Transfer data with GCM Transfer data with CBC
SSH client public key	Client authentication during SSH session negotiation	2048 - 112	Asymmetric - PSP			KAS-SSH
SSH server private key	Authentication during SSH session negotiation	2048 - 112	Asymmetric - CSP			KAS-SSH

Name	Description	Size - Strength	Type - Category	Generated By	Established By	Used By
SSH server public key	Authentication during SSH session negotiation	2048 - 112	Asymmetric - PSP			KAS-SSH
SSH-EK	SSH Encryption Key	160/256 - 160/256	Symmetric - CSP	KAS-SSH		Transfer data with GCM Transfer data with CBC
SSH-IK	SSH Integrity Key	160/256 - 160/256	Symmetric - PSP	KAS-SSH		Transfer data with CBC
SSH-SS	Shared secret computed during SSH connection	256-8192 - 112-256	Shared Secret - CSP		KAS-SSH	KAS-SSH
TLS Client key	Client authentication key	2048 - 112	Asymmetric - PSP			KAS-TLS12
TLS Server Private Key	Module authentication key	2048 - 112	Asymmetric - CSP			KAS-TLS12
TLS Server Public Key	Module authentication key	2048 - 112	Asymmetric - PSP			KAS-TLS12
GCM-IV	AES-GCM Initialization Vector	128 - -	IV - PSP	KAS-TLS12 KAS-TLS13		Transfer data with GCM
TLS-PMS	TLS 1.2 Pre-master secret	256-8192 - 112-256	Shared secret - CSP		KAS-TLS12	KAS-TLS12
TLS-SAK	TLS 1.2 session authentication key	128/256 - 128/256	Authentication - CSP	KAS-TLS12		Transfer data with CBC
TLS-SEK	TLS 1.2 Sesssion encryption key	128/256 - 128-256	Symmetric - CSP	KAS-TLS12		Transfer data with GCM Transfer data with CBC
TLS12-MS	TLS 1.2 Master Secret	384 - 384	Secret - CSP	KAS-TLS12		KAS-TLS12
TLS13-MS	TLS 1.3 Master Secret	256, 384 - 256, 384	Secret - CSP	KAS-TLS13		KAS-TLS13
TLS13-HS	TLS 1.3 Handshake secret	256, 384 - 256, 384	Secret - CSP	KAS-TLS13		KAS-TLS13
TLS13-HTS	TLS 1.3 Handshake traffic secret	256, 384 - 256, 384	Secret - CSP	KAS-TLS13		KAS-TLS13
TLS13-TK	TLS 1.3 Application traffic key	128, 256 - 128, 256	Symmetric - CSP	KAS-TLS13		Transfer data with GCM

Table 17: SSP Table 1

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
Configuration backup key		Software image:Plaintext		Service	
Configuration encryption key	Preloaded	Software image:Plaintext		Service	
Configuration integrity key	Preloaded	Software image:Plaintext		Service	
Crypto officer password	Password	Software image:Obfuscated		Service	
DH Peer Public Key	Public Key import	RAM:Plaintext		Auto	
DH Private Key		RAM:Plaintext		Auto	DH Public Key:Paired With
DH Public Key	Public Key export	RAM:Plaintext		Auto	DH Private Key:Paired With
DH-SS		RAM:Plaintext		Auto	DH Private Key:Derived from DH Peer Public Key:Derived from
DRBG Key		RAM:Plaintext		Reboot	
DRBG Output		RAM:Plaintext		Reboot	
DRBG Seed		RAM:Plaintext		Auto	
DRBG V		RAM:Plaintext		Reboot	
ECDH Peer Public Key	Public Key import	RAM:Plaintext		Auto	
ECDH Private Key		RAM:Plaintext		Auto	ECDH Public Key :Paired With
ECDH Public Key	Public Key export	RAM:Plaintext		Auto	ECDH Private Key :Paired With
ECDH-SS		RAM:Plaintext		Auto	ECDH Peer Public Key :Derived from ECDH Private Key :Derived from
Entropy input		RAM:Plaintext		Reboot	
Software update key	Preloaded	Software image:Plaintext		Service	
HA Encryption key		Software image:Encrypted		Service	
HA Password		Software image:Encrypted		Service	
IKE ECDSA Peer Public Key	Public Key import	RAM:Plaintext		Auto	
IKE ECDSA Private Key	Preloaded	Software image:Encrypted		Service	IKE ECDSA Public Key:Paired With
IKE ECDSA Public Key	Preloaded Public Key export	Software image:Encrypted		Service	IKE ECDSA Private Key:Paired With
IKE RSA Peer Public key	Public Key import	RAM:Plaintext		Auto	
IKE RSA Private Key	Preloaded	Software image:Encrypted		Service	IKE RSA Public Key:Paired With

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
IKE RSA Public Key	Preloaded Public Key export	Software image:Encrypted		Service	IKE RSA Private Key:Paired With
IKE-PSK		Software image:Encrypted		Service	
IPSec Manual Authentication key	Preloaded	Software image:Encrypted		Service	
IPsec Manual Encryption key		RAM:Plaintext		Service Reboot	
SKEYID		RAM:Plaintext		Auto	DH-SS:Derived From ECDH-SS:Derived From
SKEYID_a		RAM:Plaintext		Reboot	SKEYID:Derived From
SKEYID_e		RAM:Plaintext		Reboot	SKEYID:Derived From
SKEYSEED		RAM:Plaintext		Auto	DH-SS:Derived From ECDH-SS:Derived From
SK_a		RAM:Plaintext		Reboot	SKEYSEED:Derived From
SK_e		RAM:Plaintext		Reboot	SKEYSEED:Derived From
SSH client public key	Public Key import	RAM:Plaintext		Auto	
SSH server private key	Preloaded	Software image:Plaintext		Service	
SSH server public key	Preloaded Public Key export	Software image:Plaintext		Service	
SSH-EK		RAM:Plaintext		Reboot	SSH-SS:Derived From
SSH-IK		RAM:Plaintext		Reboot	SSH-SS:Derived From
SSH-SS		RAM:Plaintext		Auto	DH-SS:Alias of ECDH-SS:Alias of
TLS Client key	Public Key import	RAM:Plaintext		Auto	
TLS Server Private Key	Preloaded	Software image:Plaintext		Service	TLS Server Public Key:Paired With
TLS Server Public Key	Preloaded Public Key export	Software image:Plaintext		Service	TLS Server Private Key:Paired With
GCM-IV		RAM:Plaintext		Reboot	TLS12-MS:Derived From TLS13-MS:Derived From
TLS-PMS		RAM:Plaintext		Auto	DH-SS:Alias of ECDH-SS:Alias of
TLS-SAK		RAM:Plaintext		Reboot	TLS12-MS:Derived From

Name	Input - Output	Storage	Storage Duration	Zeroization	Related SSPs
TLS-SEK		RAM:Plaintext		Reboot	TLS12-MS:Derived From
TLS12-MS		RAM:Plaintext		Auto	TLS-PMS:Derived From
TLS13-MS		RAM:Plaintext		Auto	TLS13-HTS:Derived From
TLS13-HS		RAM:Plaintext		Auto	DH-SS:Derived From ECDH-SS:Derived From
TLS13-HTS		RAM:Plaintext		Auto	TLS13-HS:Derived From
TLS13-TK		RAM:Plaintext		Reboot	TLS13-MS:Derived From

Table 18: SSP Table 2

In the above table, the "Crypto officer password" is described as being stored in an obfuscated form. For this specific SSP, "obfuscated" means: hashed with the module's implementation of SHA2-256

9.5 Transitions

The module implements the following security methods that are forecasted to transition over the lifetime of the validation:

Algorithm	Uses
SHA-1	In HMAC-SHA-1 As the hash function in SSH KDF As the MGF (Mask Generation Function) in RSA signature generation
SHA2-224	In HMAC-SHA2-224 In RSA signature generation
HMAC-SHA-1	For MAC Generation As the PRF in IKEv1 KDF As the PRF in IKEv2 KDF
HMAC-SHA2-224	For MAC Generation As the PRF in IKEv1 KDF As the PRF in IKEv2 KDF

SHA-1 becomes disallowed for applying cryptographic protection in January 2031.

10 Self-Tests

When configured as described in section 11.1, the module performs pre-operational and conditional self-tests. These tests include algorithm known answer tests (KATs), pairwise consistency tests (PCTs), a software integrity test and a configuration integrity test.

If self-tests pass, module captures a "pass indicator" in this convention 'Running <algorithm name> test... passed. The following is an example of successful self-test:

Running AES test .... passed

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

In the event that any of the self-tests fail, the module logs an "error indicator" for the specific test(s). Refer to section 10.4 for details.

10.1 Pre-Operational Self-Tests

The module performs the following pre-operational self-tests:

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details
Configuration integrity	HMAC-SHA2-256	MAC	Critical Function	Console message	MAC verification
Software integrity	RSA 2048-bit, using SHA2-256	SigVer	SW/FW Integrity	Console message	Signature verification

Table 19: Pre-Operational Self-Tests

10.2 Conditional Self-Tests

The module performs the following conditional self-tests. The module contains multiple implementations of some cryptographic algorithms. The module performs self-tests of all implementations.

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
AES	AES-ECB, 128 bit	KAT	CAST	Console message	Encrypt, decrypt	On boot
AES-GCM	256-bit	KAT	CAST	Console message	Encrypt, decrypt	On boot
DH key pairs	-	PCT	PCT	Console message	Sign, verify	On key generation
DRBG	AES-256	KAT	CAST	Console message	Instantiate, generate, reseed	On boot
EC DH key pairs	-	PCT	PCT	Console message	Sign, verify, calculate public key	On key generation
ECC-SSC	P-256	KAT	CAST	Console message	Compute	On boot
ECDSA	P-256, P-384, P-521	KAT	CAST	Console message	Sign, verify	On boot
ECDSA key pairs	-	PCT	PCT	Console message	Sign verify, calculate public key	On key generation
FFC-SSC	MODP-2048	KAT	CAST	Console message	Compute	On boot

Algorithm or Test	Test Properties	Test Method	Test Type	Indicator	Details	Conditions
Software load	RSA 2048-bit with SHA-1 or SHA2-256	SigVer	SW/FW Load	Console message	-	On SW load
IKEv1 KDF	SHA2-256	KAT	CAST	Console message	Derive	On boot
IKEv2 KDF	SHA2-256	KAT	CAST	Console message	Derive	On boot
RSA	2048-bit	KAT	CAST	Console message	Sign, verify	On boot
RSA key pairs	-	PCT	PCT	Console message	Sign, verify, encrypt, decrypt	On key generation
SHA-1	-	KAT	CAST	Console message	Hash	On boot
SHA-2	SHA2-512	KAT	CAST	Console message	Hash	On boot
SSH KDF	SHA-1	KAT	CAST	Console message	Derive	On boot
TLS 1.2 KDF	SHA2-256	KAT	CAST	Console message	Derive	On boot
TLS 1.3 KDF	SHA2-256	KAT	CAST	Console message	Derive	On boot
APT	Adaptive proportion test	FDT	CAST	Module Halt	-	On generation
RCT	Repetition count test	FDT	CAST	Module Halt	-	On generation
SHA3-256 (A4977)	SHA3-256	KAT	CAST	Console Message	Hash	On boot

Table 20: Conditional Self-Tests

All cryptographic algorithm self-tests (CASTs) are performed pre-operationally – before the module transitions to the operational state (in the approved mode).

Pairwise consistency test (PCTs) are performed automatically when an asymmetric key pair is generated.

10.3 Periodic Self-Test Information

The following self-tests can be invoked by the administrator as described in section 10.5 below. The module cannot invoke these tests automatically.

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
Configuration integrity	MAC	Critical Function	-	-
Software integrity	SigVer	SW/FW Integrity	On demand	Manually

Table 21: Pre-Operational Periodic Information

Algorithm or Test	Test Method	Test Type	Period	Periodic Method
AES	KAT	CAST	On demand	Manually
AES-GCM	KAT	CAST	On demand	Manually
DH key pairs	PCT	PCT	-	-
DRBG	KAT	CAST	On demand	Manually
EC DH key pairs	PCT	PCT	-	-
ECC-SSC	KAT	CAST	-	-
ECDSA	KAT	CAST	On demand	Manually
ECDSA key pairs	PCT	PCT	-	-
FFC-SSC	KAT	CAST	-	-
Software load	SigVer	SW/FW Load	-	-
IKEv1 KDF	KAT	CAST	On demand	Manually
IKEv2 KDF	KAT	CAST	On demand	Manually
RSA	KAT	CAST	On demand	Manually
RSA key pairs	PCT	PCT	-	-
SHA-1	KAT	CAST	On demand	Manually
SHA-2	KAT	CAST	On demand	Manually
SSH KDF	KAT	CAST	On demand	Manually
TLS 1.2 KDF	KAT	CAST	On demand	Manually
TLS 1.3 KDF	KAT	CAST	On demand	Manually
APT	FDT	CAST	-	-
RCT	FDT	CAST	-	-
SHA3-256 (A4977)	KAT	CAST	On demand	Manually

Table 22: Conditional Periodic Information

10.4 Error States

Name	Description	Conditions	Recovery Method	Indicator
Error mode	All data output and cryptographic services are inhibited in the error state.	If any self-test fails	Power cycling	Self-tests failed Entering error mode... The system is going down NOW !! The system is halted

Table 23: Error States

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.

To resume operation, power cycle the VM. If the self-tests pass after the reboot, the module will resume normal operation in the approved mode. If a self-test continues to fail after rebooting, there is likely a serious problem and the VM should not be used or have network access until the problem is solved

10.5 Operator Initiation of Self-Tests

The administrator can run self-tests at any time. To run all of the tests, enter the following CLI command:

```
execute fips kat all
```

To run an individual test, enter:

```
execute fips kat <test_name>
```

To see the list of valid test names, enter:

```
execute fips kat ?
```

11 Life-Cycle Assurance

11.1 Installation, Initialization, and Startup Procedures

The FortiGate-VM virtual appliance software is shipped in a non-FIPS 140-3 compliant configuration. The following steps must be performed to put the module into a FIPS compliant configuration:

1. Download the model specific FIPS validated software image from the Fortinet Support site at https://support.fortinet.com
2. Verify the integrity of the software image
3. Install the FIPS validated software image
4. Enable the FIPS-CC mode of operation by issuing the following command from the Local Console:
5. config system fips-cc
6. set status enable
7. end
8. The Operator is required to supply a password for the admin account which will be assigned the Crypto Officer role. The supplied password must be at least 8 characters long and correctly verified, as which point the system will restart. Upon restart, the module will execute self-tests to ensure the correct initialization of the module's cryptographic functions.
9. After restarting, the Crypto Officer can confirm that the module is running in a compliant state by issuing the following command via the CLI:
10. get system status
11. The system status output will display the line:
12. FIPS-CC mode: enable

Additional information can be found in the "FIPS 140-3 and Common Criteria Compliant Operation for FortiOS 7.2" that can be found on the Fortinet technical documentation website at https://docs.fortinet.com. In addition, FIPS 140-3 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 approved 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.
- Administrative passwords are changed regularly.
- 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
- Client side RSA certificates must use 2048 bit or greater key sizes.
- For IPSec VPN tunnels using AES-GCM, Fortinet recommends the default phase 1 key lifetime of 86,400 seconds.
- If the network requires a volume-based lifetime metric, 90,000 kB is recommended.

Once the FIPS validated software has been installed and the module properly configured in the approved mode of operation, the module is running in a FIPS compliant configuration.

11.2 Administrator Guidance

- FortiOS Administration Guide

- FortiOS CLI Reference
- FortiOS Log Message Reference

11.3 Non-Administrator Guidance

None.

11.4 Design and Rules

In approved mode, remote administration via HTTP or Telnet is disabled. HTTPS, SSH or the console should be used. The approved mode of operation restricts the cipher suites used by HTTPS and SSH to a subset of the FIPS 140-3 compliant suites. Thus, the administrator does not need to take any specific actions to ensure compliance.

To use the web-based manager in approved mode, your web browser application must meet the following requirements:

- Authentication algorithm: RSA or ECDSA
- Connection security: TLS 1.2 or TLS 1.3

11.7 Additional Information

The module has been designed to enforce specific security requirements:

- Administrative passwords are at least 8 characters long.
- 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)
- Diffie-Hellman groups of less than 2048 bits are not used.

12 Mitigation of Other Attacks

The module includes a real-time Intrusion Prevention System (IPS) as well as antivirus protection, web content filtering, DNS filtering, application control and data leak prevention. Use of these capabilities is optional.

12.1 Attack List

The FortiOS IPS has two components: a signature based component for detecting attacks passing through the FortiGate 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.

12.2 Mitigation Effectiveness

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 application control can detect and take action against network traffic depending on the application generating the traffic. FortiOS application control uses the FortiGuard application control database.

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

FortiOS data leak prevention is used to prevent sensitive data from leaving your network. After sensitive data patterns are defined, data matching the patterns will either be blocked or logged and then allowed.

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

12.3 Guidance and Constraints

For complete information refer to the FortiOS Installation Guide. For the specific module in question, the FortiOS Administration Guide.



Copyright© 2026 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.