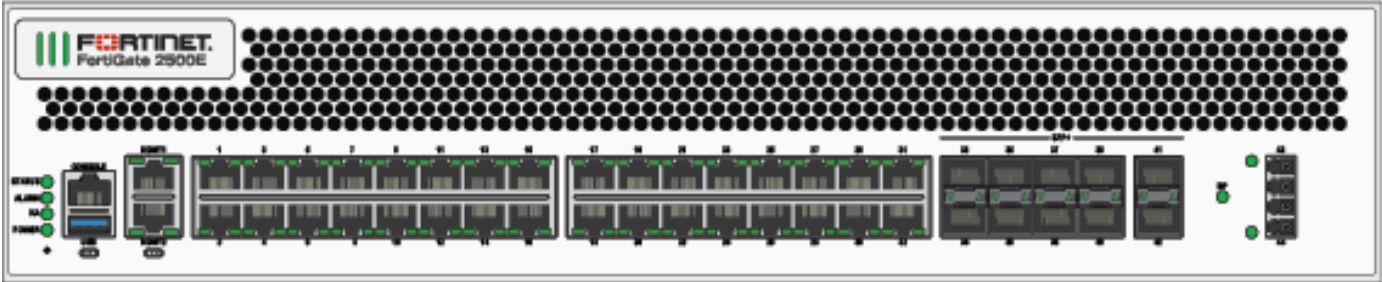


FIPS 140-2 Non-Proprietary Security Policy

FortiOS 6.2



| | |
|--|--|
| FortiOS 6.2 FIPS 140-2 Non-Proprietary Security Policy | |
| Document Version: | 3.2 |
| Publication Date: | Wednesday, September 6, 2023 |
| Description: | Documents FIPS 140-2 Level 1 Security Policy issues, compliancy and requirements for FIPS compliant operation. |
| Firmware Version: | FortiOS 6.2 build 5203 |

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://www.fortinet.com/support/contact.html>

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

Wednesday, September 6, 2023

FortiOS 6.2 FIPS 140-2 Non-Proprietary Security Policy

01-600-582610-20190913

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..... | 10 |
| Web-Based Manager..... | 10 |
| Command Line Interface..... | 11 |
| Roles, Services and Authentication..... | 11 |
| Roles..... | 11 |
| FIPS Approved Services..... | 11 |
| Non-FIPS Approved Services..... | 14 |
| Authentication..... | 14 |
| Operational Environment..... | 15 |
| Cryptographic Key Management..... | 15 |
| Random Number Generation..... | 15 |
| Entropy..... | 16 |
| Key Zeroization..... | 16 |
| Algorithms..... | 16 |
| Cryptographic Keys and Critical Security Parameters..... | 19 |
| Alternating Bypass Feature..... | 23 |
| Key Archiving..... | 23 |
| Mitigation of Other Attacks..... | 24 |
| FIPS 140-2 Compliant Operation | 25 |
| Enabling FIPS-CC Mode..... | 26 |
| Self-Tests | 27 |
| Startup and Initialization Self-tests..... | 27 |
| Conditional Self-tests..... | 27 |
| Critical Function Self-tests..... | 28 |
| Error State..... | 28 |

Overview

This document is a FIPS 140-2 Security Policy for the Fortinet FortiOS 6.2 firmware, which runs on the FortiGate family of Next Generation Firewalls. This policy describes how the FortiOS 6.2 firmware (hereafter referred to as the 'modules') meets the FIPS 140-2 security requirements and how to operate the modules in a FIPS compliant manner. This policy was created as part of the FIPS 140-2 Level 1 validation of the modules.

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 <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://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

The FortiGate family of Next Generation Firewalls spans the full range of network environments, from SOHO to service provider, offering cost effective systems for any size of application. FortiGate 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 appliances deliver a full range of network-level services — VPN, intrusion prevention, web filtering, antivirus, antispam and traffic shaping — in dedicated, easily managed platforms.

All FortiGate appliances employ Fortinet's unique FortiASIC content processing chip and the powerful, secure, FortiOS firmware achieve breakthrough price/performance. The unique, ASIC-based architecture analyzes content and behavior in real time, enabling key applications to be deployed right at the network edge where they are most effective at protecting enterprise networks. They can be easily configured to provide antivirus protection, antispam protection and content filtering in conjunction with existing firewall, VPN, and related devices, or as complete network protection systems. The modules support High Availability (HA) in both Active-Active (AA) and Active-Passive (AP) configurations.

FortiGate appliances support the IPsec industry standard for VPN, allowing VPNs to be configured between a FortiGate appliance and any client or gateway/firewall that supports IPsec VPN. FortiGate 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 | 1 |
| 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 | 2 |
| Mitigation of Other Attacks | 1 |

Module Descriptions

The modules consist of a firmware based operating system that runs exclusively on Fortinet's FortiGate product family. FortiGate units are PC-based, purpose built appliances. The FortiGate appliances are multiple chip, standalone cryptographic modules consisting of production grade components contained in a physically protected enclosure. The extent of the cryptographic boundary for the module is the outer metal chassis.

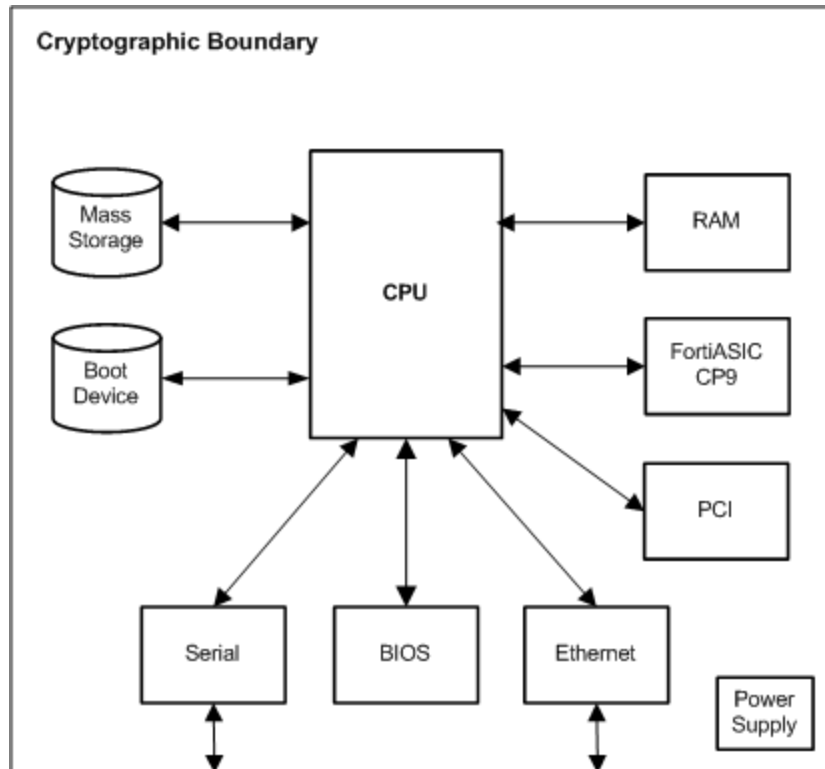


Figure 1: FortiOS physical cryptographic boundary

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

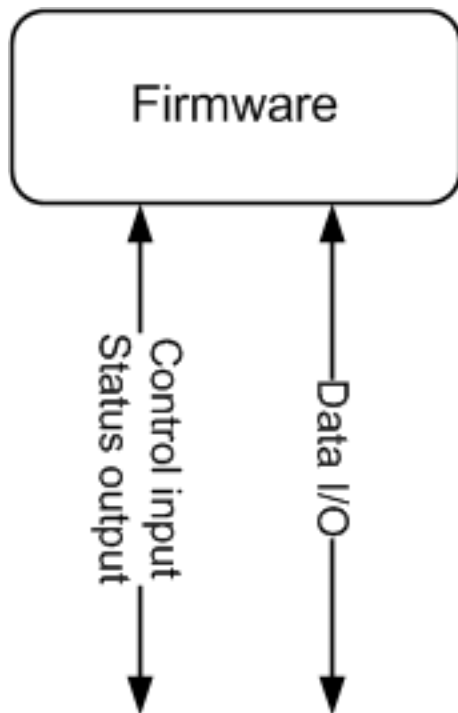


Figure 2: FortiOS logical cryptographic boundary

For the purposes of FIPS 140-2 conformance testing, the module was tested on a FortiGate-2500E appliance with Intel® Xeon® processor E5-1660v4. The extent of the physical cryptographic boundary is the FortiGate-2500E, however the validated module is the FortiOS 6.2 firmware module.

The validated firmware version is FortiOS 6.2 build 5203. Any firmware loaded that is not shown on the module certificate, is out of the scope of this validation and requires a separate FIPS 140-2 validation.

The module can also be executed on any of the following FortiGate/FortiWiFi/FortiGateRugged appliances and remain vendor affirmed FIPS-compliant.

Table 2: Vendor affirmed FIPS-compliant appliances

| | |
|----------------------------|----------------|
| FortiGateRugged-30D | FortiGate-401E |
| FortiGate/FortiWiFi-30E | FortiGate-500D |
| FortiGate/FortiWiFi-30E-MI | FortiGate-500E |
| FortiGate/FortiWiFi-30E-MN | FortiGate-501E |
| FortiGate/FortiWiFi-30E-MG | FortiGate-600D |
| FortiGateRugged-35D | FortiGate-600E |
| FortiGate/FortiWiFi-40F | FortiGate-601E |

| | |
|---|--------------------|
| FortiGate/FortiWiFi-40F-3G4G | FortiGate-900D |
| FortiGate/FortiWiFi-41F | FortiGate-1000D |
| FortiGate/FortiWiFi-41F-3G4G | FortiGate-1100E |
| FortiGate/FortiWiFi-50E | FortiGate-1100E-DC |
| FortiWifi-50E-2R | FortiGate-1101E |
| FortiGate/FortiWiFi-51E | FortiGate-1200D |
| FortiGate-52E | FortiGate-1500D/DT |
| FortiGate/FortiWiFi-60E | FortiGate-1500D-DC |
| FortiGate/FortiWiFi-60E-DSL | FortiGate-2000E |
| FortiGate/FortiWiFi-60E-DSLJ | FortiGate-2200E |
| FortiGate-60E-PoE | FortiGate-2201E |
| FortiGate/FortiWiFi-61E | FortiGate-3000D |
| FortiGate/FortiGateRugged/FortiWiFi-60F | FortiGate-3000D-DC |
| FortiGateRugged-60F-3G4G | FortiGate-3100D |
| FortiGate/FortiWiFi-61F | FortiGate-3100D-DC |
| FortiGate-80D | FortiGate-3200D |
| FortiGate-80E | FortiGate-3200D-DC |
| FortiGate-80E-PoE | FortiGate-3300E |
| FortiGate-81E | FortiGate-3301E |
| FortiGate-81E-PoE | FortiGate-3400E |
| FortiGateRugged-90D | FortiGate-3401E |
| FortiGate-90E | FortiGate-3401E-DC |
| FortiGate-91E | FortiGate-3600E |
| FortiGate-92D | FortiGate-3600E-DC |
| FortiGate-100D | FortiGate-3601E |
| FortiGate-100E | FortiGate-3700D |
| FortiGate-100EF | FortiGate-3700D-DC |
| FortiGate-101E | FortiGate-3800D |
| FortiGate-100F | FortiGate-3800D-DC |
| FortiGate-101F | FortiGate-3810D |
| FortiGate-140D | FortiGate-3810D-DC |
| FortiGate-140D-PoE | FortiGate-3815D |

| | |
|--------------------|--------------------|
| FortiGate-140E | FortiGate-3815D-DC |
| FortiGate-140E-PoE | FortiGate-3960E |
| FortiGate-200E | FortiGate-3960E-DC |
| FortiGate-201E | FortiGate-3980E |
| FortiGate-300D | FortiGate-3980E-DC |
| FortiGate-300E | FortiGate-5001D |
| FortiGate-301E | FortiGate-5001E |
| FortiGate-400D | FortiGate-5001E1 |
| FortiGate-400E | |

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.

Module Interfaces

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

Table 3: FortiOS logical interfaces and physical ports

| FIPS 140 Interface | Logical Interface | Physical Interface |
|--------------------|-----------------------|--|
| Data Input | API input parameters | Network interface, USB port |
| Data Output | API output parameters | Network Interface, USB port |
| Control Input | API function calls | Network Interface, serial interface, CP9 |
| Status Output | API return values | Network interface, serial interface |
| Power Input | 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-CC mode and is disabled.

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 FortiGate unit and the management computer. The console connection is a direct serial connection. Terminal emulation software is required on the management computer using either method. For network access, a Telnet or SSH client that supports the SSH v2.0 protocol is required (SSH v1.0 is not supported in FIPS-CC mode). Telnet access to the CLI is not allowed in FIPS-CC mode and is disabled.

Roles, Services and Authentication

Roles

When configured in FIPS-CC 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-execute 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-execute 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 module does not utilize non-compliant NIST SP-800-56Ar3 functionality in the approved mode of operation.

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

The access types are abbreviated as follows:

| | |
|-----------------------|---|
| Read Access | R |
| Write Access | W |
| Execute Access | E |

Table 4: Services available to Crypto Officers

| Service | Access | Key/CSP |
|--|--------|---|
| connect to module locally using the console port | WE | N/A |
| connect to module remotely using TLS* | WE | Diffie-Hellman Keys, EC Diffie Hellman Keys, TLS Premaster Secret, TLS Master Secret, HTTPS/TLS Server/Host Key, HTTPS/TLS Session Integrity Key, and HTTPS/TLS Session Encryption Key, DRBG v and key values, DRBG Output, DRBG Seed, NDRNG Output String, TLS Server Signatures |
| connect to module remotely using SSH* | WE | Diffie-Hellman Keys, 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, Firmware 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 firmware update | WE | Firmware Update Key |
| read log data | N/A | N/A |

| Service | Access | Key/CSP |
|---|--------|---|
| 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 |
| 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 Keys, EC Diffie-Hellman Keys SSL: HTTPS/TLS Server/Host Key, HTTPS/TLS Session Integrity Key, HTTPS/TLS Session Encryption Key |
| read/set/modify HA configuration | WE | HA Password, HA Encryption Key |
| log offloading to remote FortiAnalyzer device* | E | OFTP Client Key, Diffie-Hellman Keys, EC Diffie-Hellman Keys, TLS Premaster Secret, TLS Master Secret, HTTPS/TLS Session Integrity Key, HTTPS/TLS Session Encryption Key, HTTPS/TLS Server/Host Key, DRBG v and key values, DRBG Output, DRBG Seed, NDRNG Output String |
| generate CSR with RSA or ECDSA | WE | RSA keys, ECDSA keys |

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

| Service/CSP | Access | Key/CSP |
|---------------------------------------|--------|---|
| connect to module remotely using TLS* | WE | Diffie-Hellman Keys, EC Diffie-Hellman Keys, TLS Premaster Secret, TLS Master Secret, HTTPS/TLS Server/Host Key, HTTPS/TLS Session Integrity Key, HTTPS/TLS Session Encryption Key, DRBG v and key values, DRBG Output, DRBG Seed, NDRNG Output String, TLS Server Signatures |
| authenticate to module | WE | Network User Password |

| Service/CSP | Access | Key/CSP |
|--|--------|--|
| IPsec VPN controlled by firewall policies* | E | IPsec: IPsec Manual Authentication Key, IPsec Manual Encryption Key, IPsec Session Authentication Key, IPsec Session Encryption Key, IKE Pre-Shared Key, IKE RSA Key, IKE ECDSA Key, IKE SKEYSEED, IKE Authentication Key, IKE Key Generation Key, IKE Session Encryption Key, Diffie-Hellman Keys, EC Diffie-Hellman Keys |
| SSL VPN controlled by firewall policies* | E | Network User Password, Diffie-Hellman Keys, EC Diffie-Hellman Keys, TLS Premaster Secret, TLS Master Secret, HTTPS/TLS Server/Host Key, HTTPS/TLS Session Integrity 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 4 and 5 are considered non-approved when using the following algorithms:
 - Non-compliant-strength Diffie-Hellman

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

Authentication

The module implements identity based authentication. Crypto Officers must authenticate with a user-id and password combination to access the modules remotely or locally via the console. Remote Crypto Officer 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.

Authentication at level 3 is only applicable when identity-based authentication is enforced for the User role.

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 3,346,172,314,938,369 which is significantly lower than one in a million.

Note that Crypto Officer 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 3,346,172,314,938,369 which is less than 1/100,000.

Crypto Officer 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/(3,346,172,314,938,369/108,000) 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 2048 bit RSA key size)
- 1 in 2^{128} for the IKE ECDSA Key (based on a P-256 curve ECDSA key size)

A gigabit ethernet connection is 1,048,576,000 bits per second which is 62,914,560,000 bits per minute. An 8-byte key would have 64 bits, so there could be no more than 983,040,000 password attempts per minute. Therefore, the minimum odds of guessing the IKE Preshared key for IPsec within a one-minute period is 1 in $94^8/983,040,000$ which is less than 1 in 100,000. Similarly, for the IPsec Manual Authentication key, the minimum odds of Network Users guessing the key within a minute would be 1 in $16^{40}/393,216,000$. Guessing the IKE RSA key within a minute would be 1 in $2^{112}/561,737,143$. Guessing the IKE ECDSA key within a minute would be 1 in $2^{128}/491,520,000$.

Operational Environment

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

For the purposes of FIPS 140-2 conformance testing, the module was tested on a FortiGate-2500E unit.

Cryptographic Key Management

Random Number Generation

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

Entropy

The module uses the Fortinet's CP9 Security Processor to seed the DRBG during the modules' boot process and to periodically reseed the DRBG. The CP9 is used by default as the FortiOS entropy source - i.e. no configuration changes are required.

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 (as per section 6.4 of SP 800-90B) is applied.

Reseed Period

The RBG is seeded from the CP9 Security Processor 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).

Key Zeroization

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

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

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

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

```
execute erase-disk ?
```

Algorithms

Table 6: FIPS approved algorithms

| Algorithm | NIST Cert Number |
|--|---------------------|
| CTR DRBG (NIST SP 800-90A) with AES 256 bits | C1573 |
| AES in CBC mode (128, 192, and 256 bits) (192 bits for C1578 only) | C1549, C1575, C1578 |
| AES in GCM mode (128, 256 bits) | C1575, C1576, C1578 |
| SHA-1 | C1575, C1576, C1578 |
| SHA-224 | A1187 |
| SHA-256 | C1575, C1576, C1578 |
| SHA-384 | C1575, C1576, C1578 |
| SHA-512 | C1575, C1576, C1578 |

| Algorithm | NIST Cert Number |
|---|---------------------|
| HMAC SHA-1 | C1575, C1576, C1578 |
| HMAC SHA-256 | C1575, C1576, C1578 |
| HMAC SHA-384 | C1575, C1576, C1578 |
| HMAC SHA-512 | C1575, C1576, C1578 |
| RSA PKCS 1.5 Key Pair Generation: 2048 and 3072-bit (** C1576, C1578 only) 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 | C1576, C1578, A1252 |
| RSA PSS Signature Generation: 2048 and 3072-bit Signature Verification: 1024, 2048 and 3072-bit | A1187 |
| ECDSA Key Pair Generation: curve P-256 | C1575, C1576, C1578 |
| ECDSA Key Pair Generation: curve P-384 | C1575, C1578 |
| ECDSA Key Pair Generation: curve P-521 | C1575, C1576, C1578 |
| ECDSA Signature Generation: curves P-256, P-384 and P-521 | C1575, C1576, C1578 |
| ECDSA Signature Verification: curves P-256, P-384 and P-521 | C1575, C1576, C1578 |
| CVL (KDF SSH) - AES 128 bit-, AES 256 bit -CBC (using SHA1, SHA-256) | C1576 |
| CVL (KDF TLS 1.1 and 1.2 (using SHA-256, SHA-384)) | C1576, C1578 |
| CVL (KDF TLS 1.1 and 1.2 (using SHA-512)) | C1578 |
| CVL (KDF IKE v1 (using SHA-1, SHA2-256, SHA2-384, SHA2-512)) | C1575, C1578 |

| Algorithm | NIST Cert Number |
|---|---------------------|
| CVL (KDF IKE v2 (using SHA-1, SHA2-256, SHA2-384, SHA2-512)) | C1575, C1578 |
| CVL (ECDSA SigGen Component: Curves P-256, P-384 and P-521) | C1575, C1576, C1578 |
| KAS-ECC-SSC SP-800-56Ar3: Scheme ephemeralUnified. curves P-256, P-384, and P-521 | A1187 |
| KAS-FFC-SSC SP-800-56Ar3: Scheme dhEphem: FC | A1187 |
| CVL (KDF SNMP) - Password length: 64 - 128 | C1576 |

KTS (AES Cert. #C1549 and HMAC Cert. #C1576; key establishment methodology provides 128 or 256 bits of encryption strength). The relevant mode is AES-CBC with HMAC.

KTS (AES Cert. #C1576; key establishment methodology provides 128 or 256 bits of encryption strength). The relevant mode is AES-GCM.

KAS-ECC-SSC SP-800-56Ar3 (Cert. #A1187; provides between 128 and 256 bits of encryption strength).

KAS-FFC-SSC SP-800-56Ar3 (Cert. #A1187; provides between 112 and 196 bits of encryption strength).

For AES GCM IPsec/IKEv2, RFC 7296 is used to establish the shared secret SKEYSEED from which the AES GCM encryption keys are derived.

Table 7: FIPS allowed algorithms

| Algorithm |
|-----------------------|
| NDRNG (FortiASIC CP9) |

Table 8: Non-FIPS approved algorithms

| Algorithm |
|--|
| Diffie-Hellman is non-compliant when keys less than 2048 bits are used, since such keys do not provide the minimum required 112 bits of encryption strength. |
| 4096-bit RSA signature generation is non-compliant. |
| The following ECC curves shall not be used in the Approved mode of operation: brainpoolP224r1, brainpoolP256r1, brainpoolP384r1, brainpoolP512r1, Curve25519 and Curve448. |

Note that the IKE, SSH, SNMP, 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 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. The cipher suites implemented in the module that utilize AES-GCM are consistent with those specified in Section 3.3.1.1.2 of [SP800-52, Rev2]. 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, the key used for the AES GCM encryption or decryption shall be re-distributed.

There are algorithms, modes, and keys that have been CAVs tested but are not used by the module. Only the algorithms, modes/methods, and key lengths/curves/moduli shown in the above tables are used by the module.

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.

Table 9: 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 | By erasing the Boot device and power cycling the module |
| DRBG seed | Internally generated | SDRAM 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 | SDRAM 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 | SDRAM Plain-text | Internal state values for the DRBG 128 and 256 | By erasing the Boot device and power cycling the module |
| IPsec Manual Authentication Key | Electronic key entry | Boot device AES encrypted | Used as IPsec Session Authentication Key | By erasing the Boot device and power cycling the module |
| IPsec Manual Encryption Key | Electronic key entry | 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 | Internally generated via DH or ECDH KAS | SDRAM Plain-text | VPN traffic encryption/decryption using AES (128, 256 bit) | By erasing the Boot device and power cycling the module |

| Key or CSP | Generation | Storage | Usage | Zeroization |
|----------------------------|--|------------------------------|---|---|
| IKE SKEYSEED | Derived via KDF defined in SP800-135 (IKEv2) | SDRAM Plain-text | Used to generate IKE protocol keys | By erasing the Boot device and power cycling the module |
| IKE Pre-Shared Key | Electronic key entry | Boot device AES encrypted | Used to generate IKE protocol keys | By erasing the Boot device and power cycling the module |
| 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 Key Generation 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 (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 (signatures using P-256, P-384 and P-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 |
| Firmware Update Key | Preconfigured | Boot device Plain-text | Verification of firmware integrity when updating to new firmware versions using RSA public key (firmware load test, 2048 bit signature) | By erasing the boot device and power cycling the module |

| Key or CSP | Generation | Storage | Usage | Zeroization |
|----------------------------------|--|---------------------------|--|---|
| Firmware Integrity Key | Preconfigured | Boot device Plain-text | Verification of firmware integrity in the firmware integrity test using RSA public key (firmware integrity test, 2048 bit signature) | By erasing the boot device and power cycling the module |
| TLS Premaster Secret | Internally generated via DH or ECDH KAS | SDRAM Plain-text | HTTPS/TLS keying material | By erasing the boot device and power cycling the module |
| TLS Master Secret | Internally generated from the TLS Premaster Secret | SDRAM Plain-text | 384 bit master key used in the HTTPS/TLS protocols | 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 Integrity Key | Internally generated using DRBG | SDRAM Plain-text | HMAC SHA-1, -256 or -384 key used for HTTPS/TLS session integrity | By erasing the boot device and power cycling the module |
| TLS Server Signatures | Preconfigured | Boot device Plain-text | rsa_pkcs1 & rsa_pss_rsae signatures used in TLS | 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 |

| Key or CSP | Generation | Storage | Usage | Zeroization |
|------------------------------|---------------------------------|------------------------------|---|---|
| Crypto Officer Password | Electronic key entry | 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 bypass test | By erasing the boot device and power cycling the module |
| Configuration Encryption Key | Preconfigured | Boot device Plain-text | AES 256 bit key used to encrypt CSPs on the Boot device and in the backup configuration file (except for crypto officer passwords in the backup configuration file) | By erasing the boot device and power cycling the module |
| Configuration Backup Key | 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 | Electronic key entry | 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 | Electronic key entry | 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 |
| OFTP Client Key | Externally generated | Boot device Plain-text | RSA private key used in the OFTP/TLS protocol (key establishment, 2048 bit signature) | By erasing the boot device and power cycling the module |
| RSA Keys | Internally generated using DRBG | Boot device Plain-text | RSA Key Pair from RSA CSR generation | By erasing the boot device and power cycling the module |
| ECDSA Keys | Internally generated using DRBG | Boot device Plain-text | ECDSA Key Pair from ECDSA CSR generation | By erasing the boot device and power cycling the module |

| Key or CSP | Generation | Storage | Usage | Zeroization |
|-------------------|---|---------------------|---|---|
| Shared Secret "Z" | Computed using random inputs and predefined functions | SDRAM Plain-text | SSC Shared Secret Z for NIST SP 800-56Ar3 | By erasing the boot device and power cycling the module |

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

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.

In both cases, two independent internal actions must be taken to create a bypass firewall policies.

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, web content filtering, DNS filtering, application control and data leak prevention. 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.

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 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 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.

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

1. Download the model specific FIPS validated firmware image and checksum 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 checksum listing.
3. Install the FIPS validated firmware image from a TFTP server using the BIOS boot menu. To access the BIOS boot menu, use the console connection and press any key when the "Press any key to display the configuration menu" option is displayed during the boot process. Then select "[T]: Initiate TFTP firmware transfer" and follow the instructions to complete the installation of the firmware image.
4. Enable the FIPS-CC mode of operation as per the "Enabling FIPS-CC Mode" section.

Additional information can be found in the FortiOS 6.0 and 6.2 "FIPS 140-2 and Common Criteria Technote" 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 unit. You must ensure that:

- The FortiGate unit is configured in the FIPS-CC mode of operation.
- The FortiGate unit is installed in a secure physical location.
- Physical access to the FortiGate unit is restricted to authorized operators.
- Administrative passwords are at least 8 characters long.
- Administrative passwords are changed regularly.
- Administrator account passwords must have the following characteristics:
 - One (or more) of the characters must be capitalized
 - One (or more) of the characters must be lower case
 - One (or more) of the characters must be numeric
 - One (or more) of the 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 operation modes: NAT/Route or Transparent. NAT/Route mode applies security features between two or more different networks (for example, between a private network and the Internet). Transparent mode applies security features at any point in a network. The current operation mode is displayed on the web-based manager status page and in the output of the `get system status` CLI command.

Once the FIPS validated firmware has been installed and the module properly configured in the FIPS-CC mode of operation, the module is running in a FIPS compliant configuration. It is the responsibility of the CO to ensure the module only uses approved algorithms and services to maintain the module in a FIPS Approved mode of operation. Using any of the non-approved algorithms and services will mean that the module is operating in a non-FIPS approved mode of operation for the duration of time that the non-approved algorithm or service is being utilized. When switching back to the use of approved or allowed algorithms and services, the module is considered to be operating in the approved mode again. This is not enforced by the module itself, but by this policy. Prior to switching between modes, the CO shall ensure that all keys and CSPs are zeroized to prevent sharing of keys and CSPs between the FIPS Approved and non-FIPS mode of operation.

Enabling FIPS-CC Mode

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

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

The Operator is required to supply a password for the admin account which will be assigned to the Crypto Officer role.

The supplied password must be at least 8 characters long and correctly verified before the system will restart in FIPS-CC mode.

Upon restart, the module will execute self-tests to ensure the correct initialization of the module's cryptographic functions.

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

```
get system status
```

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

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

Self-Tests

Startup and Initialization Self-tests

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

- Firmware integrity test using RSA 2048-bit signatures
- Configuration/VPN bypass test using HMAC SHA-256
- AES (128, 256 bit), CBC mode, encrypt known answer test
- AES (128, 256 bit) CBC mode, decrypt known answer test
- AES (128, 256 bit), GCM mode, encrypt known answer test
- AES (128, 256 bit), 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 2048-bit signature generation known answer test
- RSA 2048-bit signature verification known answer test
- ECDSA pairwise consistency test
- DRBG known answer tests (as per SP 800-90A)
- Primitive-Z known answer test (KAS-FFC-SSC and KAS-ECC-SSC)
- TLS 1.1 KDF known answer test
- TLS 1.2 KDF known answer test
- IKEv1 KDF known answer test
- IKEv2 KDF known answer test
- SSH KDF 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
- Firmware load test using RSA 2048-bit 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

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.

Copyright© 2023 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.