FIPS 140-2 Non-Proprietary Security Policy

FortiAnalyzer 6.2

FortiAnalyzer 6.2 FIPS 140-2 Security Policy	
Document Version:	3.3
Publication Date:	Tuesday, November 1, 2022
Description:	Documents FIPS 140-2 Level 1 Security Policy issues, compliancy and requirements for FIPS compliant operation.
Firmware Version:	FortiAnalyzer v6.2, build9599

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

Tuesday, November 1, 2022

FortiAnalyzer 6.2 FIPS 140-2 Non-Proprietary Security Policy

05-627-0609017-20201102

TABLE OF CONTENTS

Overview

This document is a FIPS 140-2 Security Policy for Fortinet's FortiAnalyzer 6.2 firmware, which runs on the FortiAnalyzer family of security appliances. This policy describes how the FortiAnalyzer 6.2 firmware (hereafter referred to as the ‘module’) meets the FIPS 140-2 security requirements and how to operate the module in a FIPS compliant manner. This policy was created as part of the FIPS 140-2 Level 1 validation of the module.

The FortiAnalyzer family of logging, analyzing, and reporting appliances securely aggregate log data from Fortinet devices and other syslog-compatible devices. Using a comprehensive suite of customizable reports, users can filter and review records, including traffic, event, virus, attack, Web content, and email data.

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

References

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

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

- Find general product information in the product section of the Fortinet corporate website at https://www.fortinet.com/products.
- Find on-line product support for registered products in the technical support section of the Fortinet corporate website at https://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.

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

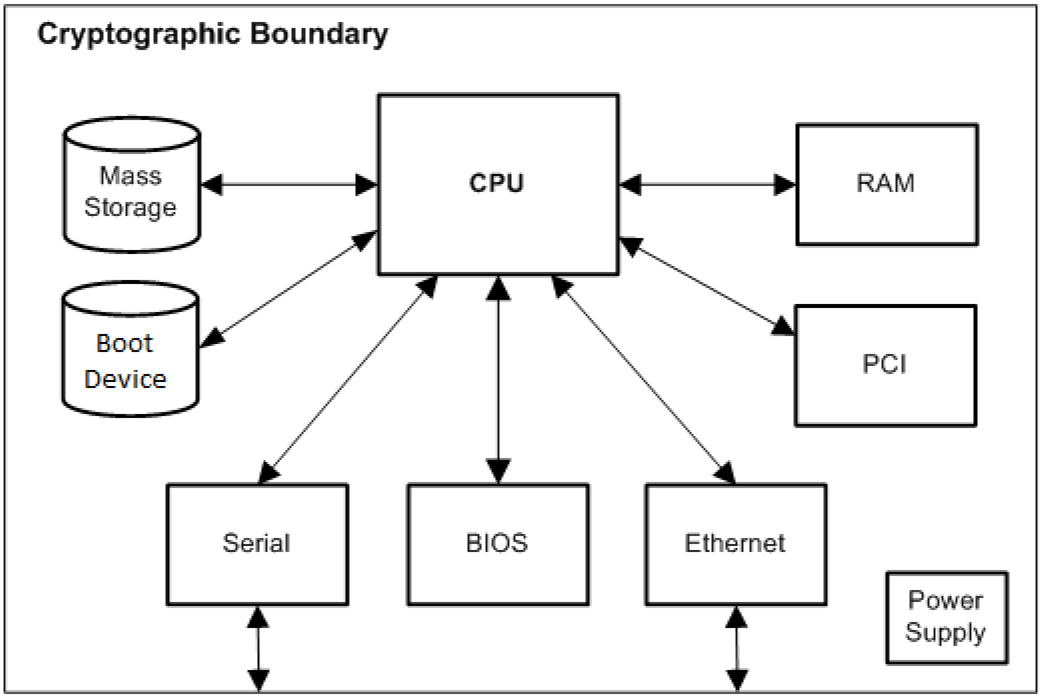
Security Requirement	Compliance Level
Cryptographic Module Specification	1
Cryptographic Module Ports and Interfaces	1
Roles, Services and Authentication	3
Finite State Model	1
Physical Security	1
Operational Environment	N/A
Cryptographic Key Management	1
EMI/EMC	1
Self-Tests	1
Design Assurance	2
Mitigation of Other Attacks	N/A

Module Descriptions

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

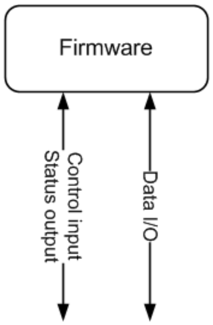
The FortiAnalyzer appliances are multiple chip, standalone cryptographic modules consisting of production grade components contained in a physically protected enclosure.

Figure 1 - FortiAnalyzer 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 FortiAnalyzer appliance.

Figure 2 - FortiAnalyzer logical cryptographic boundary



For the purposes of FIPS 140-2 conformance testing, the module was tested on a FortiAnalyzer-3500G appliance and used an entropy token (Araneus Alea II) as the entropy source.

The extent of the cryptographic boundary for the module is the outer metal chassis.

The validated firmware version is FortiAnalyzer v6.2, build9599. Any firmware version that is not shown on the module certificate is out of scope of this validation and requires a separate FIPS 140-2 validation.

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

Table 2: Vendor affirmed FIPS-compliant appliances

FortiAnalyzer-200D	FortiAnalyzer-2000E
FortiAnalyzer-200F	FortiAnalyzer-3000E
FortiAnalyzer-300F	FortiAnalyzer-3000F
FortiAnalyzer-400E	FortiAnalyzer-3000G
FortiAnalyzer-800F	FortiAnalyzer-3500E
FortiAnalyzer-1000E	FortiAnalyzer-3500F
FortiAnalyzer-1000F	FortiAnalyzer-3700F
	FortiAnalyer-3900E

Module Interfaces

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

Table 3: FortiAnalyzer logical interfaces and physical ports

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

Command Line Interface

The FortiAnalyzer 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 FortiAnalyzer 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 access to all of the Module’s administrative services. The initial Crypto Officer can create additional Crypto Officer operator accounts by giving the account super user access permissions.

The Module provides a Network User role. Network users have read/write access to a restricted set of the Module's administrative services. Network User accounts are created by Crypto Officers and do not have super user access permissions.

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
authenticate to module*	WE	Crypto Officer Password, Diffie-Hellman Key, EC Diffie-Hellman Key, HTTPS/TLS Server/Host Key, HTTPS/TLS Session Encryption Key, SSH Server/Host Key, SSH Session Authentication Key, SSH Session Encryption Key, DRBG v and key values, DRBG Output, DRBG Seed, Entropy String, TLS Server Signatures
show system status	R	N/A
show FIPS-CC mode enabled/disabled (console/CLI only)	R	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	All keys stored in Flash RAM
execute FIPS-CC on-demand self-tests (console only)	E	Configuration Integrity Key, Firmware Integrity Key

Service	Access	Key/CSP
add/delete crypto officer and network users	WE	Crypto Officer Password, Network User Password
set/reset crypto officer and network user passwords	WE	Crypto Officer Password, Network User Password
modify user preferences	RWE	N/A
backup/restore configuration file	RWE	Configuration Encryption Key, Configuration Backup Key
read/set/delete/modify module configuration	RW	N/A
execute firmware update	WE	Firmware Update Key
read log data	R	N/A
delete log data (console/CLI only)	W	N/A
format log disk (console/CLI only)	W	N/A
execute system diagnostics (console/CLI only)	E	N/A
offload logs to remote FAZ server	R	OFTP Client Key
read/set/delete/modify local and remote log configuration	RW	OFTP Client Key
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	Network User Password, Diffie-Hellman Key, EC Diffie-Hellman Key, HTTPS/TLS Server/Host Key, HTTPS/TLS Session Encryption Key, DRBG v and key values, DRBG Output, DRBG Seed, Entropy String, TLS Server Signatures
access to log data	RE	N/A
modify user preferences	E	N/A

*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.

Non-FIPS Approved Services

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

- Configuration backups using password protection
- RADIUS authentication

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

Authentication

The module uses identity based authentication. By default, operators and users authenticate with a username and password combination to access the module. The username/password can be stored in the local database or in a remote LDAP database. Remote operator authentication is done over HTTPS (TLS) or SSH. Local operator authentication is done over the console connection. Remote user authentication is done over HTTPS (TLS). Password entry is obfuscated using asterisks. The feedback mechanism does not provide information that could be used to guess or determine the authentication data.

Note that the network user’s username and password are not stored on the module. The module operates as a logging device. User authentication is done over HTTPS which uses the underlying TLS protocol to protect user data between the client and the module and the module and the back end server during the authentication process. Network User access to the module can be based on policy and authentication by IP address.

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 1uppercase 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.

Operational Environment

The module constitutes the entire firmware operating system for a FortiAnalyzer unit and can only be installed and run on a FortiAnalyzer 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 FortiAnalyzer-3500G unit with Intel® Xeon® Gold 5118 processor.

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 an entropy token (Araneus Alea II) to seed the DRBG during the modules' boot process and to periodically reseed the DRBG. The entropy token is not included in the boundary of the module and therefore no assurance can be made for the correct operation of the entropy token nor is there a guarantee of stated entropy.

Entropy Strength

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

Reseed Period

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

Key Zeroization

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

All keys and CSPs are zeroized by erasing the module's boot device and then power cycling the FortiAnalyzer 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 FortiAnalyzer module. Executing the following command will output a list of the available internal disks:

```
execute erase-disk ?
```

Algorithms

Table 6: FIPS approved algorithms

Algorithm	NIST Certificate Number
CTR DRBG (NIST SP 800-90A) with AES 256 bits	C1985
AES in CBC mode (128, 256 bits)	C1908
AES in GCM mode (128, 256 bits)	C2013
AES in CTR mode (128, 192, 256 bits)	A1062
SHA-1	C2013
SHA-256	C2013
SHA-384	C2013
SHA-512	C2013
HMAC SHA-1	C2013
HMAC SHA-256	C2013
HMAC SHA-384	C2013
HMAC SHA-512	C2013
RSA PKCS 1.5 186-4 Signature Generation: 2048 and 3072 bit 186-4 Signature Verification: 1024, 2048 and 3072 bit Key Pair Generation: 2048 and 3072 bit (***Cert. #A1062 only) For legacy use, the module supports 1024-bit RSA keys and SHA-1 for signature verification	C2013, A1062
CVL (SSH) - AES 128 bit, AES 256 bit -CTR (using SHA-1, SHA-256)	C2013
CVL (TLS 1.0, 1.1 and 1.2 (using SHA-256))	C2013
ECDSA Key Pair Generation: curves P-256, P-384, and P-521	A1062

Algorithm	NIST Certificate Number
ECDSA Key Pair Verification: curves P-256, P-384, and P-521	A1062
ECDSA Signature Generation: curves P-256, P-384, and P-521	A1062
ECDSA Signature Verification: curves P-256, P-384, and P-521	A1062
KAS-ECC-SSC Sp-800-56Ar3: Scheme: ephemeralUnified. curves P-256, P-384, and P-521	A1062
KAS-FFC-SSC Sp-800-56Ar3: Scheme: dhEphem: FB, FC	A1062

KTS (AES Cert. #C1908 and HMAC Cert. #C2013; key establishment methodology provides 128 or 256 bits of encryption strength). The relevant mode is AES-CBC with HMAC.

KTS (AES Cert. #C2013; key establishment methodology provides 128 or 256 bits of encryption strength). The relevant mode is AES-GCM.

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

SHA-1 is utilized for legacy digital signature verification and for hashing.

KAS-FFC (Cert. #A1062, CVL Cert. #C2013; key agreement; key establishment methodology provides between 112 bits and 196 bits of encryption strength. **

KAS-ECC (Cert. #A1062, CVL Cert. #C2013; key agreement; key establishment methodology provides between 128 bits and 256 bits of encryption strength. **

** - Please note that the inclusion of CVL Cert. #C2013 in the above entries represents the TLS KDF portion of the KAS implementation.

Table 7: Non-FIPS approved algorithms

Algorithm
SNMP (The SNMP KDF has not undergone CAVP testing in accordance with NIST SP 800-135, Rev1, and thus SNMP shall not be used in the Approved mode. Any use of SNMP will cause the module to operate in a non-Approved mode.)
4096-bit RSA signature generation is non-compliant.

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

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

For TLS, The GCM implementation meets Option 1 of IG A.5: it is used in a manner compliant with SP 800-52 and in accordance with RFC 5246 for TLS key establishment. The AES GCM IV generation is in compliance with RFC 5288 and shall only be used for the TLS protocol version 1.2 to be compliant with FIPS140-2 IG A.5, Option 1 ("TLS protocol IV generation"); thus, those 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.

SSHv2 is compliant with RFC 4252, 4253, and 5647. SSHv2 is compliant with Option 1 of IG A.5.

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.

The following ECC curves shall not be used in the Approved mode of operation: brainpoolP224r1, brainpoolP256r1, brainpoolP384r1, brainpoolP512r1, Curve25519 and Curve448.

Cryptographic Keys and Critical Security Parameters

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

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

Key or CSP	Generation	Storage	Usage	Zeroization
Entropy String	Externally generated (entropy token)	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 external entropy token)	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	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	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	By erasing the Boot device and power cycling the module

Key or CSP	Generation	Storage	Usage	Zeroization
Firmware Update Key	Preconfigured	Boot device Plain-text	Verification of firmware integrity when updating to new firmware versions using RSA public key (firmware load test, 2048 bit signature)	By erasing the boot device and power cycling the module
Firmware Integrity Key	Preconfigured	Boot device Plain-text	Verification of firmware integrity in the firmware integrity test using RSA public key (firmware integrity test, 2048 bit signature)	By erasing the boot device and power cycling the module
HTTPS/TLS Server/Host Key	Preconfigured	Boot device Plain-text	RSA private key used in the HTTPS/TLS protocols (key establishment, 2048 or 3072 bit)	By erasing the boot device and power cycling the module
HTTPS/TLS Session Encryption Key	Internally generated via DH KAS	SDRAM Plain-text	AES key used for HTTPS/TLS session encryption	By erasing the boot device and power cycling the module
TLS Server Signatures	Preconfigured	Boot device Plain-text	RSA signatures used in TLS	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 KAS	SDRAM Plain-text	AES (128, 256 bit) key used for SSH session encryption	By erasing the boot device and power cycling the module
Crypto Officer Password	Electronic key entry	Boot device SHA-1 hash	8-32 character password used to authenticate operator access to the module	By erasing the boot device and power cycling the module

Key or CSP	Generation	Storage	Usage	Zeroization
Configuration Integrity Key	Preconfigured	Boot device Plain-text	HMAC SHA-256 hash used for configuration integrity test	By erasing the boot device and power cycling the module
Configuration Encryption Key	Preconfigured	Boot device Plain-text	AES 256-bit key used to encrypt CSPs on the Boot device and in the backup configuration file (except for crypto officer passwords in the backup configuration file)	By erasing the boot device and power cycling the module
Configuration Backup Key	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 module
Network User Password	Electronic key entry	Boot device SHA-256 hash	8-32 character password used to authenticate network access to the module	By erasing the boot device and power cycling the module
OFTP Client Key	Externally generated	Boot device AES encrypted	RSA private key used in the OFTP protocol (key establishment, 2048 or 3072 bit)	By erasing the boot device and power cycling the module
TLS Premaster Secret	Internally generated via DH KAS	SDRAM Plain-text	HTTPS/TLS keyring material	By erasing the boot device and power cycling the module
TLS Master Secret	Internally generated from the TLS Premaster Secret	SDRAM Plain-text	256-bit master key used in the HTTPS/TLS protocols	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, or -256 key used for HTTPS/TLS session integrity	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

Key or CSP	Generation	Storage	Usage	Zeroization
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
Shared Secret "Z"	Internally generated using DRBG	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.

Key Archiving

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

Mitigation of Other Attacks

The module does not mitigate against any other attacks.

FIPS 140-2 Compliant Operation

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

1. Download the model specific FIPS validated firmware image from the Fortinet Support site at https://support.fortinet.com/
2. Verify the integrity of the firmware image by validating the MD5 checksum or SHA-512 checksum of the image
3. Install the FIPS validated firmware image
4. Install the entropy token (Araneus Alea II, available at https://www.araneus.fi/products/alea2/en)
5. Enable the FIPS-CC mode of operation

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 FortiAnalyzer unit. You must ensure that:

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

Once the FIPS validated firmware has been installed and the module properly configured in the FIPS-CC mode of operation, the module is running in a FIPS compliant configuration. It is the responsibility of the CO to ensure the module only uses approved algorithms and services to maintain the module in a FIPS-CC Approved mode of operation. Using any of the non-approved algorithms and services 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
   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 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
- AES, CBC mode, encrypt known answer test
- AES, CBC mode, decrypt known answer test
- AES, GCM mode, encrypt known answer test
- AES, GCM mode, decrypt known answer test
- HMAC SHA-1 known answer test
- SHA-1 known answer test (tested as part of HMAC SHA-1 known answer test)
- HMAC SHA-256 known answer test
- SHA-256 known answer test (tested as part of HMAC SHA-256 known answer test)
- HMAC SHA-384 known answer test
- SHA-384 known answer test (tested as part of HMAC SHA-384 known answer test)
- HMAC SHA-512 known answer test
- SHA-512 known answer test (tested as part of HMAC SHA-512 known answer test)
- RSA signature generation known answer test
- RSA signature verification known answer test
- DRBG known answer test
- Primitive-Z known answer test (KAS-FFC-SSC and KAS-ECC-SSC)
- ECDSA pairwise consistency test
- TLS 1.1 KDF known answer test
- TLS 1.2 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 entropy input test
- Continuous DRBG test
- RSA pairwise consistency test

- ECDSA pairwise consistency test
- Configuration integrity test using HMAC SHA-256
- Firmware load test using RSA signatures

Critical Function Self-tests

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

- Instantiate test
- Generate test
- Reseed test

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, where "XXX" is a description of the cause of the problem, such as "AES self-test":

```
FIPS error: "XXX" failed.
Entering error mode...
```

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

High Performance Network Security