# McAfee NGFW Cryptographic Kernel Module

# FIPS 140-2 Non-Proprietary Security Policy

Version 2.1

Last Update: 2014-10-24

Prepared by:
atsec information security Corp.
9130 Jollyville Road, Suite 260
Austin, TX 78759
www.atsec.com

# 1. Introduction

This document is a non-proprietary FIPS 140-2 Security Policy for the McAfee NGFW Cryptographic Kernel Module. The current version of this module is 2.0. An earlier version of this module has gone through FIPS 140-2 validation under certificate #1991. This document contains a specification of the rules under which the module must operate and describes how this module meets the requirements as specified in the Federal Information Processing Standards Publication (FIPS PUB) 140-2 for a Security Level 1 multi-chip standalone software module.

## 1.1. Purpose of the Security Policy

There are three major reasons that a security policy is required:

- For FIPS 140-2 validation,
- Allows individuals and organizations to determine whether the cryptographic module, as implemented, satisfies the stated security policy, and
- Describes the capabilities, protection, and access rights provided by the cryptographic module, allowing individuals and organizations to determine whether it will meet their security requirements.

## 1.2. Target Audience

This document is intended to be part of the package of documents that are submitted for FIPS 140-2 validation. It is intended for the following people:

- Developers working on the release
- FIPS 140-2 testing lab
- Cryptographic Module Validation Program (CMVP)
- Consumers

# 2. Cryptographic Module Specification

This document is the non-proprietary security policy for the McAfee NGFW Cryptographic Kernel Module, and was prepared as part of the requirements for conformance to FIPS 140-2, Level 1.

The following section describes the module and how it complies with the FIPS 140-2 standard in each of the required areas.

## 2.1. Description of Module

The McAfee NGFW Cryptographic Kernel Module is a multi-chip standalone, software-only module that provides general-purpose cryptographic algorithms for McAfee applications. The binary of the module and the integrity check file are qcl_fips.ko and checksums.fips. Assembly language optimizations are used in the cryptographic module implementation. The module contains the following cryptographic functionality:

- Cryptographic hash functions
- Message authentication code functions
- Symmetric key encryption and decryption

The following table shows the overview of the security level for each of the eleven sections of the validation.

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

*Table 1: Security Levels*

The module has been tested on the following platforms:

| Manufacturer | Model | O/S & Ver. | AES-NI |
|---|---|---|---|
| McAfee | MIL-320 | Debian GNU/Linux 6.0-based distribution (single-user mode) | Not Support |

| | | | |
|---|---|---|---|
| McAfee | 5206 | Debian GNU/Linux 6.0-based distribution (single-user mode) | With AES-NI |
| McAfee | 3206 | Debian GNU/Linux 6.0-based distribution (single-user mode) | With and Without AES-NI |
| McAfee | 3202 | Debian GNU/Linux 6.0-based distribution (single-user mode) | With and Without AES-NI |
| McAfee | 1402 | Debian GNU/Linux 6.0-based distribution (single-user mode) | With AES-NI |
| McAfee | 1065 | Debian GNU/Linux 6.0-based distribution (single-user mode) | With AES-NI |
| McAfee | 1035 | Debian GNU/Linux 6.0-based distribution (single-user mode) | With AES-NI |

*Table 2: Tested Platforms*

## 2.2. Description of Approved Mode

The cryptographic module supports only a FIPS 140-2 approved mode. The cryptographic module is initialized and set in the FIPS 140-2 approved mode when the kernel module is loaded.

The calling application can call the ssh_crypto_get_certification_mode() function to confirm the current mode of operation. It returns SSH_CRYPTO_CERTIFICATION_FIPS_140_2 to indicate that the module is indeed in the FIPS 140-2-approved mode.

The module supports the following approved functions:

- AES: ECB, CBC, OFB, CFB128 and GCM modes
- Triple-DES: ECB, CBC, OFB and CFB64 modes
- SHS: hashing
- HMAC: message integrity

## 2.3. Cryptographic Module Boundary

### 2.3.1. Software Block Diagram

The logical boundary of the module is the binary code of the Cryptographic Kernel Module itself, which is indicated by the “Cryptographic Boundary” rectangle as illustrated in the diagram below.

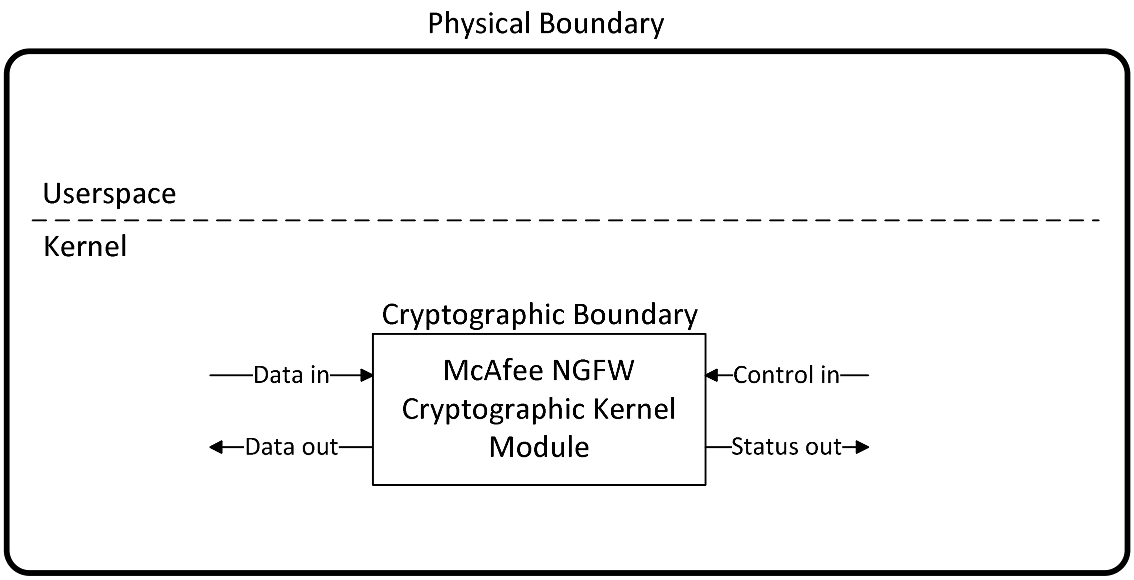
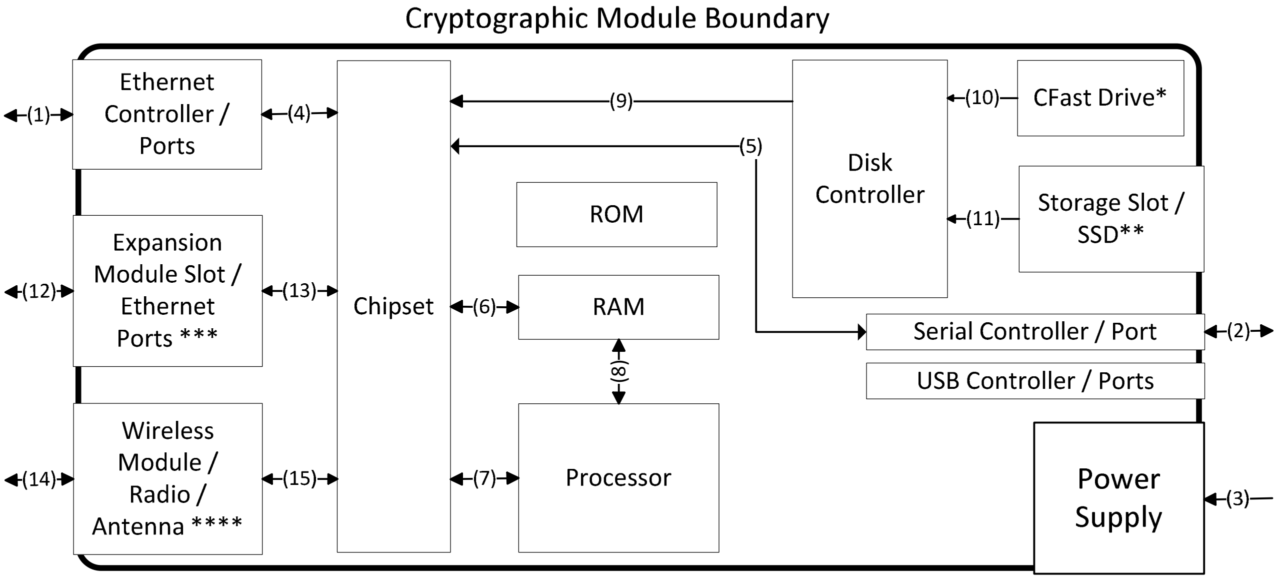


*Figure 1: Software Block Diagram*

### 2.3.2. Hardware Block Diagram

The physical boundary of the module is the enclosure of the appliance that the module is running on. The module was tested on seven separate appliances, all of which are general purpose computers. The hardware block diagram below depicts all test appliances and their internal components and ports (processor, SSD, USB, Ethernet, etc.).



1, 2, 4, 5, 6, 7, 8, 12, 13, 14 and 15: Data in, data out, control in, status out
3: Power in
9, 10 and 11: Control in

*) MIL-320, 1035, 1065
**) 1402, 3202, 3206, 5206
***) 1035, 1065, 1402, 3202, 3206, 5206
****) MIL-320 only

*Figure 2: Hardware Block Diagram*

# 3. Cryptographic Module Ports and Interfaces

| FIPS Interface | Physical Ports | Logical Ports |
|---|---|---|
| Data Input | Ethernet ports, serial port, wireless radio | API input parameters |
| Data Output | Ethernet ports, serial port, wireless radio | API output parameters and return values |
| Control Input | Serial port, Ethernet ports, wireless radio | API functions, API input parameters |
| Status Output | Serial port, Ethernet ports, wireless radio | API return values, console, kernel ring buffer |
| Power Input | PC power supply port | N/A |

*Table 3: Ports and Interfaces*

# 4. Roles, Services, and Authentication

## 4.1. Roles

The module implements both a User and a Cryptographic Officer (CO) role. The module does not allow concurrent operators.

The User role assumes to perform Approved algorithms operations. The CO role assumes to perform module installation and initialization.

The User and CO roles are implicitly assumed by the entity accessing services implemented by the module. No further authentication is required. The CO can initialize the module.

## 4.2. Services

| Service | Roles | | CSP | Modes | FIPS Approved? Cert # (if applicable) | Access | Notes/API function |
|---|---|---|---|---|---|---|---|
| | User | CO | | | | | |
| **Symmetric Algorithms** | | | | | | | |
| AES encryption and decryption | ✓ | | 128, 192, 256 bit keys | ECB, CBC, OFB, CFB128 | Yes<br>Certs #2914, 2915, 2916, 2917, 2918, 2919, 2920, 2921 | RWX | FIPS 197<br>ssh_cipher_allocate<br>ssh_cipher_free<br>ssh_cipher_get_block_length<br>ssh_cipher_get_iv<br>ssh_cipher_get_iv_length<br>ssh_cipher_get_key_length<br>ssh_cipher_get_max_key_length<br>ssh_cipher_get_min_key_length<br>ssh_cipher_get_supported<br>ssh_cipher_has_fixed_key_length<br>ssh_cipher_is_fips_approved<br>ssh_cipher_name<br>sh_cipher_set_iv<br>ssh_cipher_supported<br>ssh_cipher_transfor |

| Service | Roles | | CSP | Modes | FIPS Approved? Cert # (if applicable) | Access | Notes/API function |
|---|---|---|---|---|---|---|---|
| | User | CO | | | | | |
| | | | | | | | m<br>ssh_cipher_transform_remaining<br>ssh_cipher_transform_with_iv<br>ssh_cipher_get_block_len |
| AES-GCM authenticated encryption and decryption | ✓ | | 128, 192, 256 bit keys | GCM | Yes<br>Certs #2914, 2915, 2916, 2917, 2918, 2919, 2920, 2921 | RWX | SP 800-38D<br>ssh_cipher_allocate<br>ssh_cipher_free<br>ssh_cipher_get_block_length<br>ssh_cipher_get_iv<br>ssh_cipher_get_iv_length<br>ssh_cipher_get_key_length<br>ssh_cipher_get_max_key_length<br>ssh_cipher_get_min_key_length<br>ssh_cipher_get_supported<br>ssh_cipher_has_fixed_key_length<br>ssh_cipher_is_fips_approved<br>ssh_cipher_name<br>sh_cipher_set_iv<br>ssh_cipher_supported<br>ssh_cipher_transform<br>ssh_cipher_transform_remaining<br>ssh_cipher_transform_with_iv<br>ssh_cipher_get_block_len |

| Service | Roles | | CSP | Modes | FIPS Approved? Cert # (if applicable) | Access | Notes/API function |
|---|---|---|---|---|---|---|---|
| | User | CO | | | | | |
| | | | | | | | ssh_cipher_is_auth_cipher<br>ssh_cipher_auth_reset<br>ssh_cipher_auth_update<br>ssh_cipher_auth_final<br>ssh_cipher_auth_digest_length<br>ssh_cipher_is_auth<br>ssh_cipher_generate_iv_ctr<br>ssh_cipher_auth_digest_len |

| Service | Roles | | CSP | Modes | FIPS Approved? Cert # (if applicable) | Access | Notes/API function |
|---|---|---|---|---|---|---|---|
| | User | CO | | | | | |
| Triple-DES encryption and decryption | ✓ | | 168 bit keys | ECB, CBC, OFB, CFB64 | Yes<br>Certs #1729, 1730, 1731, 1732, 1733, 1734 | RWX | SP 800-67<br>ssh_cipher_allocate<br>ssh_cipher_free<br>ssh_cipher_get_block_length<br>ssh_cipher_get_iv<br>ssh_cipher_get_iv_length<br>ssh_cipher_get_key_length<br>ssh_cipher_get_max_key_length<br>ssh_cipher_get_min_key_length<br>ssh_cipher_get_supported<br>ssh_cipher_has_fixed_key_length<br>ssh_cipher_is_fips_approved<br>ssh_cipher_name<br>sh_cipher_set_iv<br>ssh_cipher_supported<br>ssh_cipher_transform<br>ssh_cipher_transform_remaining<br>ssh_cipher_transform_with_iv<br>ssh_cipher_get_block_len |
| **Hash Functions** | | | | | | | |
| SHA-1<br>SHA-224<br>SHA-256<br>SHA-384<br>SHA-512 | ✓ | | | N/A | Yes<br>Certs #2452, 2453, 2454, 2455, | RX | FIPS 180-4<br>ssh_hash_allocate<br>ssh_hash_digest_length<br>ssh_hash_final |

| Service | Roles | | CSP | Modes | FIPS Approved? Cert # (if applicable) | Access | Notes/API function |
|---|---|---|---|---|---|---|---|
| | User | CO | | | | | |
| | | | | | 2456, 2457 | | ssh_hash_free<br>ssh_hash_get_supported<br>ssh_hash_input_block_size<br>ssh_hash_is_fips_approved<br>ssh_hash_name<br>ssh_hash_reset<br>ssh_hash_supported<br>ssh_hash_update |
| **Message Authentication Codes (MACs)** | | | | | | | |
| HMAC-SHA1<br>HMAC-SHA224<br>HMAC-SHA-256<br>HMAC-SHA-384<br>HMAC-SHA-512 | ✓ | | At least 112 bits HMAC key | N/A | Yes<br>Certs #1843, 1844, 1845, 1846, 1847, 1848 | RWX | FIPS 198-1<br>ssh_mac_allocate<br>ssh_mac_final<br>ssh_mac_free<br>ssh_mac_get_block_length<br>ssh_mac_get_max_key_length<br>ssh_mac_get_min_key_length<br>ssh_mac_get_supported<br>ssh_mac_is_fips_approved<br>ssh_mac_length<br>ssh_mac_name<br>ssh_mac_reset<br>ssh_mac_supported<br>ssh_mac_update |

| Service | Roles | | CSP | Modes | FIPS Approved? Cert # (if applicable) | Access | Notes/API function |
|---|---|---|---|---|---|---|---|
| | User | CO | | | | | |
| **Management** | | | | | | | |
| Installation | | ✓ | N/A | N/A | N/A | N/A | Please refer to section 11.3 “Cryptographic Officer Guidance” for secure installation of the module. |
| Initialization | | ✓ | N/A | N/A | N/A | RX | ssh_crypto_library_initialize<br>sg_crypto_register_error_callback |
| Mode management | | ✓ | N/A | N/A | N/A | RX | ssh_crypto_get_certification_mode<br>ssh_crypto_set_certification_mode |
| Uninitialization | | ✓ | N/A | N/A | N/A | RX | ssh_crypto_free<br>ssh_crypto_library_uninitialize |
| External crypto registration | | ✓ | N/A | N/A | N/A | RX | The external crypto registration is not supported on the tested Stonesoft platforms. The functions below return SG_CRYPTO_REGISTER_NOT_SUPPORTED.<br>sg_cipher_external_register<br>sg_cipher_external_unregister<br>sg_hash_external_register<br>sg_hash_external_unregister<br>sg_mac_external_register<br>sg_mac_external_u |

| Service | Roles | | CSP | Modes | FIPS Approved? Cert # (if applicable) | Access | Notes/API function |
|---|---|---|---|---|---|---|---|
| | User | CO | | | | | |
| | | | | | | | nregister<br>sg_ciphermac_external_register<br>sg_ciphermac_external_unregister |
| **Status** | | | | | | | |
| Query status | ✓ | ✓ | N/A | N/A | N/A | RX | ssh_crypto_library_get_status<br>ssh_crypto_library_get_version<br>ssh_crypto_status_message |
| **Self-tests** | | | | | | | |
| Perform self-tests | ✓ | ✓ | N/A | N/A | N/A | RX | ssh_crypto_library_self_tests |

*Table 4: Services*

## 4.3. Operator Authentication

There is no operator authentication; assumption of role is implicit by action.

## 4.4. Mechanism and Strength of Authentication

No authentication is required at Security Level 1; authentication is implicit by assumption of the role.

# 5. Finite State Machine

The following diagram represents the states and transitions of the cryptographic module.

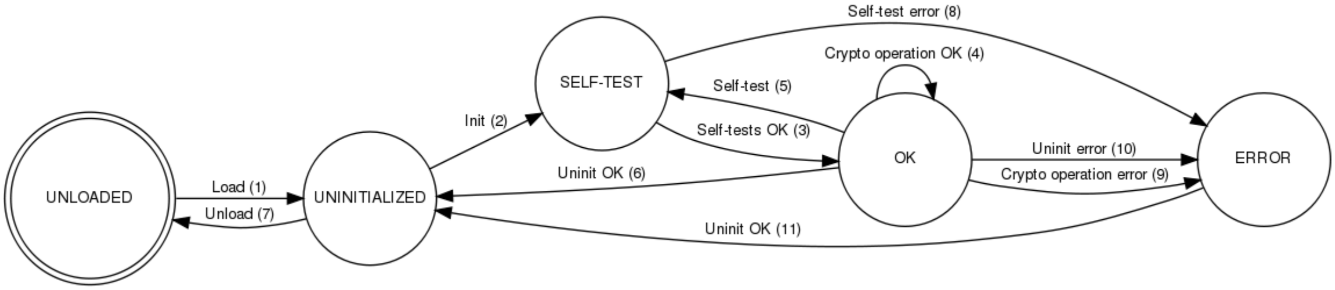


*Figure 3: Cryptographic Module Finite State Machine*

The state model contains the following states:

- UNLOADED: The start state of the cryptographic module is UNLOADED. The module is in this state until the shared library is loaded and linked to the application. Cryptographic operations are not available while in this state.
- UNINITIALIZED: The module is in the UNINITIALIZED state after it has been loaded but not yet initialized, or it has been successfully uninitialized. Cryptographic operations are not available while in this state.
- SELF-TEST: The module performs power-up self-tests during initialization or on-demand. Cryptographic operations are not available while in this state.
- OK: The cryptographic operations are available and are performed.
- ERROR: A self-test, a cryptographic operation or uninitialization has failed. An error indicator is output by the module.

The state transitions are as follows:

1. The loadable kernel module is loaded.
2. The cryptographic module is initialized using the ssh_crypto_library_initialize function. The function is called automatically when the cryptographic kernel module is loaded.
3. The self-tests succeed.
4. A cryptographic operation is performed successfully.
5. Self-tests are performed using the ssh_crypto_library_self_tests function.
6. The cryptographic module is uninitialized using the ssh_crypto_library_uninitialize function.
7. The shared library or the loadable kernel module is unloaded.
8. The self-tests fail.
9. A conditional test fails during a cryptographic operation.
10. The module uninitialization fails because cryptographic objects are still referenced.
11. Cryptographic objects are no longer in use and the module uninitialization succeeds. This transition also occurs automatically when the power-up self-tests fail during the module initialization.

# 6. Physical Security

The cryptographic module is tested on the McAfee MIL-320, 5206, 3206, 3202, 1402, 1065 and 1035 appliances that consist of production-grade components with standard passivation and a production-grade enclosure.

# 7. Operational Environment

This module will operate in a modifiable operational environment per the FIPS 140-2 definition. The module operates on the McAfee NGFW Debian GNU/Linux based hardened operating system that is set in the FIPS 140-2 compatible mode of operation. Login to the operating system is disabled and only the preinstalled McAfee application is running on the system. Therefore the operational environment is considered non-modifiable. The application that uses the cryptographic module is also the single user of the module.

# 8. Cryptographic Key Management

Keys are established externally. CSPs can be accessed only using the API. The operating system protects the memory and the address space of the process from unauthorized access.

| Name | Auth Role | Generation | Size and Type | Output | Storage | Zeroization |
|---|---|---|---|---|---|---|
| AES symmetric keys | User | External, electronic entry | 128-, 192-, 256-bit Symmetric key | N/A | Plaintext in memory | API call, power off |
| Triple-DES symmetric keys | User | External, electronic entry | 168-bit Symmetric key | N/A | Plaintext in memory | API call, power off |
| HMAC key | User | External, electronic entry | At least 112 bits Symmetric key | N/A | Plaintext in memory | API call, power off |
| HMAC key for module integrity check | User, Crypto Officer | Manufacturer | 128 bits HMAC-SHA-256 key | N/A | In module binary | Zeroization is not required per FIPS IG 7.4 |

*Table 6: Key Management*

## 8.1. Key Entry and Output

The cryptographic module supports electronic entry of symmetric keys and HMAC keys via API parameters. The application using the cryptographic module can pass secret keys to the module in plaintext within the physical boundary. The keys are associated with the calling application and are all ephemeral. Both User and Crypto Officer can enter, use and destroy symmetric keys. The module does not output any keys or CSPs.

## 8.2. Key Storage

The keys and CSPs are stored in plaintext in memory. The module does not provide persistent storage of keys.

## 8.3. Zeroization Procedure

The stored keys and CSPs are zeroized when the application calls the appropriate API function: ssh_cipher_free and ssh_mac_free. It is the calling application responsibility to call the zeroization API function to zeroize the keys and CSPs. Temporary key material is zeroized automatically by the module when no longer needed. All keys and CSPs can be zeroized by powering off the module and performing a system restore operation by the operational environment.

# 9. Electromagnetic Interference/Electromagnetic Compatibility (EMI/EMC)

All seven test platforms as shown in Table 2 are compliant to 47 CFR FCC Part 15, Subpart B, Class A (Business use).

# 10. Self-Tests

## 10.1. Power-Up Tests

The power-up self-tests are executed automatically when the cryptographic module is loaded. The kernel module initialization function returns 0 (SSH_CRYPTO_OK) when the power-up self-tests have succeeded. If the power-up self-tests fail, the cryptographic module outputs an error indicator and enters an error state. The computer will need to be restarted in order for the cryptographic module to return to an operational state. No further operations are allowed when the module is in an error state.

Self-tests are performed on-demand when the user calls the ssh_crypto_library_self_tests function.

Cryptographic algorithm tests (Known Answer Tests):

- AES encryption and decryption tested separately
- Triple-DES encryption and decryption tested separately
- HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512
- SHA-1, SHA-256, SHA-512

Cryptographic self-test error messages:

- Cipher algorithm test failed during self test.
- Mac algorithm test failed during self test.
- Hash algorithm test failed during self test.
- The checksum of the library is incorrect. Integrity has been compromised.

It is the application’s responsibility to reboot the appliance to recover the module from the error state. The library will not cause the rebooting of the appliance.

## 10.2. Integrity Check

The cryptographic module uses the HMAC-SHA-256 message authentication code of the module binary for the integrity tests. The module reads the module binary file, computes the HMAC-SHA-256 MAC of the file content and compares it to the known correct MAC that is input to the module when it is loaded.

# 11. Design Assurance

## 11.1. Configuration Management

Git and Lotus Notes are used for configuration management of the cryptographic module.

## 11.2. Delivery and Operation

The cryptographic module is never released as source code. It is delivered as part of the McAfee NGFW software (formerly Stonesoft Security Engine). The FIPS-compatible McAfee NGFW software image is downloaded from the McAfee website. The McAfee NGFW software is also preinstalled on McAfee NGFW appliances (see *Table 2: Tested Platforms*). Product information for the appliances is available at the McAfee website: http://www.mcafee.com/us/products/next-generation-firewall.aspx

### 11.2.1. Downloading a FIPS 140-2-compatible engine version

A FIPS-compatible version of the McAfee NGFW software is downloaded as follows:

1. Go to the McAfee NGFW Downloads page at https://my.stonesoft.com/download.do.
2. Enter the Proof-of-License (POL) or Proof-of-Serial (POS) code in the License Identification field and click **Submit**.
3. Click **McAfee NGFW downloads**. The McAfee NGFW Downloads page opens.
4. Download the .zip installation file.
5. Verify the SHA checksum. The correct checksum is shown on the download page.

## 11.3. Cryptographic Officer Guidance

### 11.3.1. Installation

The cryptographic module is delivered as part of the McAfee NGFW software. To run the cryptographic module on a McAfee NGFW appliance, the NGFW software is set to a FIPS-compatible operating mode.

#### 11.3.1.1 Upgrading appliances to the FIPS 140-2-compatible engine version

McAfee NGFW appliances are delivered with the most recent engine software preinstalled. The engine software must be upgraded to the FIPS 140-2-compatible engine version before entering FIPS-compatible operating mode. This is necessary even if the same version was installed previously, because the file system checksum is stored during the upgrade process.

To upgrade to the FIPS 140-2-compatible engine version:

1. Save the FIPS 140-2-compatible engine upgrade zip file in the root directory of a USB memory stick. Note – The engine upgrade zip file must be in the root directory of the media.
2. Boot up the appliance. The Engine Configuration Wizard starts.
3. Select **Upgrade**. The Select Source Media dialog opens.
4. Select **USB Memory**. The upgrade starts.

5. Select **OK**. The engine reboots and the Engine Configuration Wizard starts with the engine image verification dialog shown. Select **Calculate**. The file system checksum is calculated and displayed below the checksum from the engine image zip file.
6. Verify that the calculated checksum is identical to the checksum from the zip file.
7. Select **OK**. The engine reboots.
8. Check the engine version to make sure that the certified version is loaded.

Continue as instructed in **Configuring the engine,** below.

#### 11.3.1.2 Configuring the engine

To configure the engine:

1. Start the Engine Configuration Wizard as instructed in the **Configuring the Engine in the Engine Configuration Wizard** section of the *McAfee NGFW Installation Guide*.
2. Configure the Operating System settings as instructed in the **Configuring the Operating System Settings** section of the *McAfee NGFW Installation Guide*. Select **Restricted FIPS-compatible operating mode**. The SSH (Secure Shell) daemon and root password options are automatically disabled in the Engine Configuration Wizard.
3. Configure the network interfaces according to your environment as instructed in the **Configuring the Network Interfaces** section of the *McAfee NGFW Installation Guide*.
4. Contact the Management Server as instructed in the **Contacting the Management Server** section of the *McAfee NGFW Installation Guide*. Enter node IP address manually is selected by default and other IP address options are disabled when FIPS-compatible operating mode is enabled.

The engine restarts.

#### 11.3.1.3 Verifying activation of FIPS 140-2-compatible operating mode

Restricted FIPS 140-2-compatible operating mode must be enabled during the initial configuration of the appliance. The following steps describe how to verify that FIPS 140-2-compatible operating mode has been activated.

To verify activation of FIPS 140-2-compatible operating mode:

1. Verify that the following messages are displayed on the console when the engine restarts:

```
FIPS: rootfs integrity check OK
```

(displayed after the root file system integrity test has been executed successfully)

```
FIPS power-up tests succeeded
```

(displayed after the FIPS 140-2 power-up tests have been executed successfully)

2. Continue as instructed in the **After Successful Management Server Contact** section of the *McAfee NGFW Installation Guide*.

Note – If the engine does not enter FIPS-compatible operating mode even though it is configured to do so, or if the power-up tests fail (a power-up test error message is displayed or the success message is not displayed), the appliance must be reset to factory settings and reinstalled as instructed in **Recovering from a FIPS 140-2 self-test failure**.

#### 11.3.1.4 Resetting the appliance to factory settings

Resetting the appliance to factory settings is not part of the normal installation procedure. There is no need to reset the appliance to factory settings before starting to use it for the first time. These instructions can be used to reset the appliance to factory settings when necessary, such as when initial configuration has been completed without enabling the Restricted FIPS-compatible operating mode, during use, or when the appliance is being removed from use.

To reset the appliance to factory settings:

1. Reboot the appliance and select **System restore options** from the boot menu. McAfee NGFW System Restore starts.
2. Enter 2 for **Advanced data removal options**.
3. Enter one of the following options:
   - 1 for **1 pass overwrite**
   - 8 for a **Custom** number of overwrite passes

If you selected **Custom**, enter the number of overwrite passes. A larger number of overwrites is more secure, but it may take a considerable amount of time depending on the appliance storage capacity.

#### 11.3.1.5 Recovering from a FIPS 140-2 self-test failure

If the FIPS 140-2 power-up self-tests fail, or the engine does not enter FIPS-compatible operating mode, the appliance must be reset to factory settings and reinstalled according to these instructions. Begin by Resetting the appliance to factory settings.

To recover from a FIPS 140-2 self-test failure:

1. Reset the appliance to factory settings as instructed in **Resetting the appliance to factory settings**.
2. Repeat the engine version upgrade as instructed in **Upgrading appliances to the FIPS 140-2-compatible engine version**.
3. Configure the firewall engine and enable FIPS-compatible operating mode as instructed in **Configuring the engine**.
4. Verify that FIPS-compatible operating mode is activated as instructed in **Verifying activation of FIPS 140-2-compatible operating mode**.

### 11.3.2. Initialization

The cryptographic module is initialized by loading the kernel module before any cryptographic functionality is available. The kernel module is loaded as follows:

```
# modprobe –f <module name> msg_digest=<hash value> file_path=<module path> \
  file_size=<module size>
```

<module name> is the name of the kernel module

<hash value> is the known SHA-256 hash value for the integrity check

<module path> is the pathname of the kernel module binary

<module size> is the size of the kernel module binary

The operation is performed automatically by the McAfee NGFW software.

## 11.4. User Guidance

### 11.4.1. AES GCM

If the module's power is lost and then restored, the key used for the AES GCM encryption/decryption shall be redistributed.

### 11.4.2. Zeroization

When a cryptographic key is no longer used, the key must be zeroized and freed using the ssh_cipher_free and ssh_mac_free functions for symmetric key encryption/decryption and message authentication keys, respectively.

# 12. Mitigation of Other Attacks

No other attacks are mitigated.

# 13. Glossary and Abbreviations

| | |
|---|---|
| **AES** | Advanced Encryption Specification |
| **API** | Application Programming Interface |
| **CAVP** | Cryptographic Algorithm Validation Program |
| **CBC** | Cipher Block Chaining |
| **CFB** | Cipher Feedback |
| **CMT** | Cryptographic Module Testing |
| **CMVP** | Cryptographic Module Validation Program |
| **CO** | Cryptographic Officer |
| **CSP** | Critical Security Parameter |
| **CTR** | Counter |
| **CVT** | Component Verification Testing |
| **DES** | Data Encryption Standard |
| **ECB** | Electronic Codebook |
| **EMC** | Electromagnetic Compatibility |
| **EMI** | Electromagnetic Interference |
| **FCC** | Federal Communications Commission |
| **FIPS** | Federal Information Processing Standards |
| **FSM** | Finite State Model |
| **GCM** | Galois Counter Mode |
| **HMAC** | Hash Message Authentication Code |
| **MAC** | Message Authentication Code |
| **NIST** | National Institute of Science and Technology |
| **NVLAP** | National Voluntary Laboratory Accreditation Program |
| **OFB** | Output Feedback |
| **O/S** | Operating System |
| **POL** | Proof-of-License |
| **POS** | Proof-of-Serial |
| **SHA** | Secure Hash Algorithm |
| **SHS** | Secure Hash Standard |
| **UI** | User Interface |

# 14. References

[1] FIPS 140-2 Standard: http://csrc.nist.gov/groups/STM/cmvp/standards.html

[2] FIPS 140-2 Implementation Guidance: http://csrc.nist.gov/groups/STM/cmvp/standards.html

[3] FIPS 140-2 Derived Test Requirements: http://csrc.nist.gov/groups/STM/cmvp/standards.html

[4] FIPS 197 Advanced Encryption Standard: http://csrc.nist.gov/publications/PubsFIPS.html

[5] FIPS 180-4 Secure Hash Standard: http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

[6] FIPS 198-1 The Keyed-Hash Message Authentication Code (HMAC): http://csrc.nist.gov/publications/PubsFIPS.html

[7] SP 800-67 Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher: http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf

[8] SP 800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC: http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf