McAfee, Inc.

McAfee Vulnerability Manager Cryptographic Module

Software Version: 1.0

FIPS 140-2 Non-Proprietary Security Policy

FIPS Security Level: 1
Document Version: 0.7

Prepared for:

McAfee, Inc.
2821 Mission College Blvd.
Santa Clara, CA 95054
United States of America

Phone: +1 (408) 988-3832
Email: info@mcafee.com
http://www.mcafee.com

Prepared by:

Corsec Security, Inc.
13135 Lee Jackson Memorial Hwy., Suite 220
Fairfax, VA 22033
United States of America

Phone: +1 (703) 267-6050
Email: info@corsec.com
http://www.corsec.com

Table of Contents

Table of Figures

List of Tables

1 Introduction

1.1 Purpose

This is a non-proprietary Cryptographic Module Security Policy for the McAfee Vulnerability Manager Cryptographic Module (Software Version: 1.0) from McAfee, Inc. This Security Policy describes how the McAfee Vulnerability Manager Cryptographic Module meets the security requirements of Federal Information Processing Standards (FIPS) Publication 140-2, which details the U.S. and Canadian Government requirements for cryptographic modules. More information about the FIPS 140-2 standard and validation program is available on the National Institute of Standards and Technology (NIST) and the Communications Security Establishment Canada (CSEC) Cryptographic Module Validation Program (CMVP) website at http://csrc.nist.gov/groups/STM/cmvp.

This document also describes how to run the module in a secure FIPS-Approved mode of operation. This policy was prepared as part of the Level 1 FIPS 140-2 validation of the module. The McAfee Vulnerability Manager Cryptographic Module is referred to in this document as VMCM, the cryptographic module or the module.

1.2 References

This document deals only with operations and capabilities of the module in the technical terms of a FIPS 140-2 cryptographic module security policy. More information is available on the module from the following sources:

- The McAfee website (http://www.mcafee.com) contains information on the full line of products from McAfee.
- The CMVP website (http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm) contains contact information for individuals to answer technical or sales-related questions for the module.

1.3 Document Organization

The Security Policy document is one document in a FIPS 140-2 Submission Package. In addition to this document, the Submission Package contains:

- Vendor Evidence document
- Finite State Model document
- Other supporting documentation as additional references

This Security Policy and the other validation submission documentation were produced by Corsec Security, Inc. under contract to McAfee. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Submission Package is proprietary to McAfee and is releasable only under appropriate non-disclosure agreements. For access to these documents, please contact McAfee.

2 McAfee VMCM

2.1 Overview

McAfee® Vulnerability Manager (also called "VM") is an agentless comprehensive asset-scanning solution capable of detecting security vulnerabilities and non-compliant configurations across millions of network-based assets. VM aggregates results from priority-based vulnerability scans into flexible reports that incorporate a patented FoundScore risk formulae to prioritize results based on vulnerability, asset criticality, and severity.

VM includes up-to-date templates that assist large-scale enterprises in achieving compliance requirements such as SOX[1], PCI-DSS[2], HIPPA[3], and FISMA[4] by automating required vulnerability scans. The agent-less scans allow VM to detect new assets attached to the network and check for vulnerabilities without having to deploy or rely on preinstalled software. VM incorporates dozens of protocols in order to perform penetration testing, as well as authenticated and non-credentialed scans throughout the organization. Scans are conducted on operating system policies, database management systems, registry keys, file and drive permissions, running services, and more in order to identify threats to infrastructure security, data theft, malware and virus infestations, and more. The deployment of VM components in various ways allows organizations to centralize scanning resources at the core, or design multi-tiered, decentralized solutions allowing VM to gain awareness of every asset from servers and workstations, to printers, smartphones, and dynamic or portable assets across enclave boundaries.

Additionally, VM fully integrates with other products from McAfee including ePolicy Orchestrator as well as McAfee Global Threat Intelligence servers in order to leverage the knowledge gained by McAfee Labs researchers from the aggregation of threats detected by millions of sensors around the world, providing VM with up-to-the-minute signatures and patterns for all types of vulnerabilities.

2.1.1 Vulnerability Manager Architecture Overview

McAfee VM is a modular system comprised of many components that can be deployed in any number of configurations. Regardless of whether all components execute on a single platform or are distributed across many machines, they communicate using networking protocols. This design allows VM deployments to grow with an enterprise, and scale into very large, distributed networks.

The software is divided into five major components which represent the distinct functionality of the VM Server:

- Enterprise Manager – Uses Microsoft Internet Information Services (IIS) to provide authorized users with access to Vulnerability Manager 7.5 through their Web browsers. It allows them to manage and run Vulnerability Manager 7.5 from anywhere on the network. Access is protected by user identification and authentication. Secure Socket Layer (SSL) can be set up through the Web server to provide encrypted communications to browsers.

- Database – Is the data repository for the Vulnerability Manager system. It uses Microsoft SQL Server to store everything from scan settings and results to user accounts and scan engine settings. It contains all of the information needed to track organizations and workgroups, manage users and groups, run scans, and generate reports.

- API Server – Provides the communication between the enterprise manager and the database.

[1] SOX – Sarbanes-Oxley Act of 2002
[2] PCI-DSS – Payment Card Industry – Data Security Standard
[3] HIPAA – Health Insurance Portability and Accountability Act of 1996
[4] FISMA – Federal Information Security Management Act of 2002

- Scan Controller – Provides the communication to the scan engines. One or more scan controller can control multiple scan engines.

- Scan Engine – The server that scans the network. Depending on the logistics and size of the network, one or more scan engines may be required to scan the network.

The components of version 7.5 of VM are designed to run on Windows Server 2008 R2, and all network communication between the various VM components takes place over encrypted channels created through the use of both a McAfee-proprietary cryptographic library called McAfee Vulnerability Manager Cryptographic Module as well as the collection of libraries made available by the Microsoft Cryptography API[5]: Next Generation (CNG).

The McAfee VMCM software is written in C/C++, and compiled to run on Windows Server 2008 R2 (64-bit). The McAfee VMCM is validated at the following FIPS 140-2 Section levels listed in Table 1 below.

Table 1 – Security Level Per FIPS 140-2 Section

Section	Section Title	Level
1	Cryptographic Module Specification	1
2	Cryptographic Module Ports and Interfaces	1
3	Roles, Services, and Authentication	1
4	Finite State Model	1
5	Physical Security	N/A[6]
6	Operational Environment	1
7	Cryptographic Key Management	1
8	EMI/EMC[7]	1
9	Self-tests	1
10	Design Assurance	3
11	Mitigation of Other Attacks	N/A

2.2 Module Specification

The McAfee Vulnerability Manager Cryptographic Module is a software module (Software Version: 1.0) with a multi-chip standalone embodiment. The overall security level of the module is 1. The cryptographic boundary of the module consists of McAfee VMCM library as shown by the red-colored dotted line in Figure 1 and blue-colored dotted line in Figure 2. It is designed to execute on a host General Purpose Computer (GPC) hardware platform running Windows Server 2008 R2.

The module implements the FIPS-Approved algorithms listed in Table 2 below.

[5] API – Application Programming Interface
[6] N/A – Not Applicable
[7] EMI/EMC – Electromagnetic Interference / Electromagnetic Compatibility

Table 2 – FIPS-Approved Algorithm Implementations

Algorithm	Certificate Number
Symmetric Key Algorithm	
AES[8] – CBC[9] mode (128/192/256 bits)	2176
Triple-DES[10] – CBC mode (keying option 1 and 2[11])	1378
Asymmetric Key Algorithm	
RSA[12] (ANSI[13] X9.31) – key generation (3072 bits); signature verification (3072 bits)	1122
RSA (PKCS[14] #1.5) – signature verification (3072 bits)	1122
RSA (PSS[15]) signature verification (3072 bits)	1122
Secure Hashing Algorithm (SHA)	
SHA-1, SHA-512	1888
Message Authentication Code (MAC)	
HMAC using SHA-1 and SHA-512	1332
Pseudo Random Number Generation (PRNG)	
ANSI X9.31 Appendix A.2.4 PRNG	1102
Key Establishment Method	
RSA encrypt/decrypt[16] (3072 bits)	Non-compliant (Allowed in FIPS mode for key transport[17])

***NOTE**: The following security functions have been deemed "deprecated"or "legacy-use" by NIST. Please refer to NIST Special Publication 800-131A for specific guidance on transitions to the use of stronger cryptographic keys and more robust algorithms.*

- *ANSI X9.31 PRNG is **deprecated** through 2015, **disallowed** after 2015.*
- *RSA signature verification with SHA-1 is **legacy-use** after 2010.*

The module also employs the following non-compliant algorithms:

- 1024-bit Diffie Hellman (DH) key agreement
- 3072-bit RSA signature generation using SHA-1

2.2.1 Physical Cryptographic Boundary

As a software cryptographic module, there are no physical protection mechanisms implemented. Therefore, the module must rely on the physical characteristics of the host system. The physical boundary of the cryptographic module is defined by the hard enclosure around the host system on which it executes.

[8] AES – Advanced Encryption Standard

[9] CBC – Cipher-Block Chaining

[10] DES – Data Encryption Standard

[11] To use the two-key Triple-DES algorithm to encrypt data in an Approved mode of operation, the module operator shall ensure that the same two-key Triple-DES key is not used for encrypting data with more than 2^{20} blocks of plaintext data.

[12] RSA – Rivest Shamir Adleman

[13] ANSI – American National Standards Institute

[14] PKCS – Public-Key Cryptography Standards

[15] PSS – Probabilistic Signature Scheme

[16] Caveat: RSA (key wrapping; key establishment methodology provides 128 bits of encryption strength)

[17] The module implements RSA encrypt/decrypt, which is non-Approved. However, a calling application may use this to implement a key transport scheme, which is allowed for use in FIPS mode.

The module supports the physical interfaces of host system. These interfaces include the integrated circuits of the system board, processor, network adapters, memory, hard disk, device case, power supply, and fans. See Figure 1 for a standard GPC block diagram.

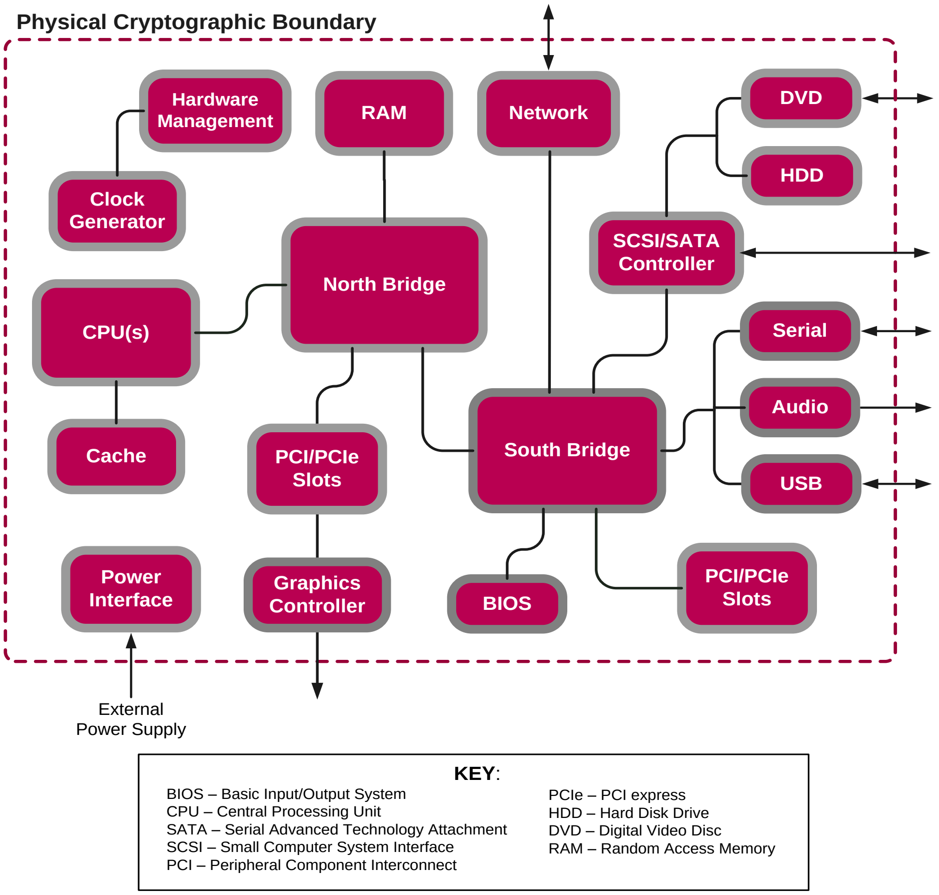

Figure 1 – GPC Block Diagram

2.2.2 Logical Cryptographic Boundary

Figure 2 shows a logical block diagram of the module, where "Calling Application" represents any other McAfee-developed software/firmware component loaded on the appliance that employs the module's services. The module's logical cryptographic boundary (also illustrated in Figure 2) encompasses all functionality provided by the module as described in this document.

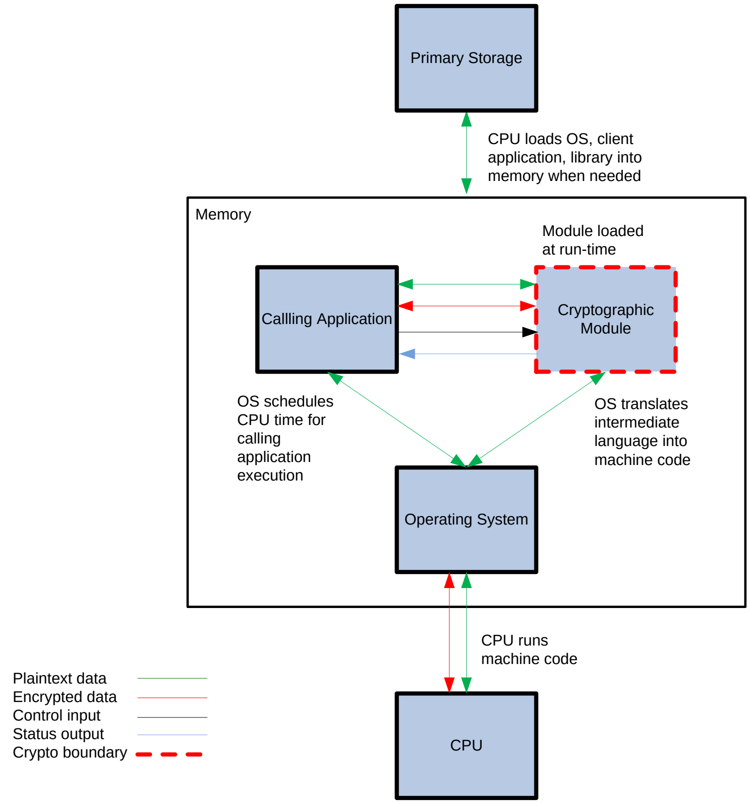
Primary Storage
CPU loads OS, client application, library into memory when needed
Memory
Module loaded at run-time
Calling Application
Cryptographic Module
OS schedules CPU time for calling application execution
OS translates intermediate language into machine code
Operating System
CPU runs machine code
CPU
Plaintext data
Encrypted data
Control input
Status output
Crypto boundary

Figure 2 – McAfee VMCM Logical Cryptographic Boundary

The cryptographic module is a shared library that provides cryptographic and secure communication services to the various McAfee-developed components of the Vulnerability Manager. In this document, those components will be referred as the calling application. The module is used by the calling application to provide encryption/decryption, hash verification, hashing, cryptographic key generation, random bit generation, and message authentication functions.

2.3 Module Interfaces

The module’s physical ports can be categorized into the following logical interfaces defined by FIPS 140-2:

- Data input
- Data output
- Control input
- Status output

As a software module, the module has no physical characteristics. The module’s physical and electrical characteristics, manual controls, and physical indicators are those of the host system. The mapping of the module’s logical interfaces in the software to FIPS 140-2 logical interfaces is described in Table 3 below.

Table 3 – FIPS 140-2 Logical Interface Mappings

FIPS 140-2 Interface	Physical Interface	Module Interface
Data Input	Network/Serial/USB port, DVD/CD, Keyboard, and Mouse	Function calls that accept, as their arguments, data to be used or processed by the module

FIPS 140-2 Interface	Physical Interface	Module Interface
Data Output	Network/Serial/USB port, DVD/CD, Graphics/Video port, and Audio	Arguments for a function that specify where the result of the function is stored
Control Input	Network/Serial/USB port, Keyboard and Mouse, Power button	Function calls utilized to initiate the module and the function calls used to control the operation of the module
Status Output	Network/Serial/USB port, Graphics/Video, LED indicators, and Audio	Return values for function calls; module-generated error messages
Power Input	Power plug/adapter, Power Switch	Not Applicable

2.4 Roles and Services

There are two roles in the module that operators may assume: a Crypto Officer role and User role. The Crypto Officer is responsible for managing the module and monitoring the module's status, while the User accesses the services implemented by the module. The available functions are utilized to provide or perform the cryptographic services.

The various services offered by the module are described in Table 4 and Table 5. Together, these lists make up the entirety of services offered by the module when running in its Approved mode of operation. The Critical Security Parameters (CSPs) used by each service are also listed. Please note that the keys and CSPs listed in the tables use the following notation to indicate the type of access required:

- R – Read: The keys and CSPs are read.
- W – Write: The keys and CSPs are established, generated, modified, or zeroized.
- X – Execute: The keys and CSPs are used within an Approved or Allowed security function or authentication mechanism.

2.4.1 Crypto Officer Role

The Crypto Officer (CO) role is responsible for zeroizing keys and CSPs, executing self-tests, and monitoring status. Descriptions of the services available to the Crypto Officer role are provided in Table 4.

Table 4 – Crypto Officer Services

Service	Description	Input	Output	CSP and Type of Access
Initialize module	Performs integrity check and power-up self-tests	API call parameters, mode	Status	Integrity check HMAC[18] key – X
Show status	Returns the current mode of the module	API call parameters; Reboot command or cycling power	Status	None
Run self-tests on demand	Performs power-up self-tests	None	Status	Integrity check HMAC key – X
Zeroize keys	Zeroizes and de-allocates memory containing sensitive data	Reboot command or cycling power	None	AES key – W TDES[19] key – W HMAC key – W RSA private/public key – W DH components – W

[18] HMAC – (Keyed-) Hash Message Authentication Code

2.4.2 User Role

The User role can utilize the module's cryptographic functionalities. Descriptions of the services available to the User role are provided in Table 5.

Table 5 – User Services

Service	Description	Input	Output	CSP and Type of Access
Generate random number (ANSI X9.31)	Returns the specified number of random bits to calling application	API call parameters	Status, random bits	ANSI X9.31 RNG seed – RX ANSI X9.31 RNG eed key – RX
Generate message digest (SHS[20])	Compute and return a message digest using SHS algorithms	API call parameters, message	Status, hash	None
Generate keyed hash (HMAC)	Compute and return a message authentication code using HMAC SHA-1	API call parameters, key, message	Status, hash	HMAC key – RX
Symmetric encryption	Encrypt plaintext using supplied key and algorithm specification (Triple-DES or AES)	API call parameters, key, plaintext	Status, ciphertext	AES key – RX TDES key – RX
Symmetric decryption	Decrypt ciphertext using supplied key and algorithm specification (Triple-DES or AES)	API call parameters, key, ciphertext	Status, plaintext	AES key – RX TDES key – RX
Generate symmetric key	Generate and return the specified type of symmetric key (Triple-DES or AES)	API call parameters	Status, key pair	AES key – W TDES key – W
Generate asymmetric key pair	Generate and return the specified type of asymmetric key pair (RSA)	API call parameters	Status, key pair	RSA private/public key – W
Asymmetric encryption	Encrypt plaintext using RSA public key (used for key transport)	API call parameters, key, plaintext	Status, ciphertext	RSA public key – RX
Asymmetric decryption	Decrypt ciphertext using RSA private key (used for key transport)	API call parameters, key, ciphertext	Status, plaintext	RSA private key – RX
DH key agreement	Perform key agreement using Diffie-Hellman algorithm	API call parameter	Status, key components	DH components – W
Signature Verification	Verify the signature on the supplied message using the specified key and algorithm (RSA)	API call parameters, key, signature, message	Status	RSA public key – RX

[19] TDES – Triple Data Encryption Standard
[20] SHS – Secure Hash Standard

2.4.3 Authentication

The module does not support any authentication mechanism. Operators of the module implicitly assume a role based on the service of the module being invoked. Since all services offered by the module can only be used by either the Crypto Officer or the User, the roles are mutually exclusive. Thus, when the operator invokes a Crypto Officer role service, he implicitly assumes the Crypto Officer role. When the operator invokes a User role service, he implicitly assumes the User role.

2.5 Physical Security

Since this is a software module, the module relies on the target platform (a GPC appliance or customer-owned hardware platform) to provide the mechanisms necessary to meet FIPS 140-2 physical security requirements. All components of the target platform will be made of production-grade materials, and all integrated circuits coated with commercial standard passivation.

2.6 Operational Environment

The McAfee VMCM was tested and found compliant with the applicable FIPS 140-2 requirements when running on the following operational environment:

- Intel Xeon running 64-bit Windows Server 2008 R2 (when operating in single user mode)

When compiled from the same unmodified source code, the module maintains its FIPS 140-2 compliance when running on the following supported operating platform:

- Intel Celeron running 64-bit Windows Server 2008 R2 (when operating in single user mode)

All cryptographic keys and CSPs are under the control of the operating system (OS), which protects the keys and CSPs against unauthorized disclosure, modification, and substitution. The module only allows access to keys and CSPs through its well-defined APIs. The module performs a Software Integrity Test using a FIPS-Approved message authentication code (HMAC SHA-1).

2.7 Cryptographic Key Management

The module supports the critical security parameters listed below in Table 6. Please note that the “Input” and “Output” columns in Table 6 are in reference to the module’s logical boundary.

Table 6 – Cryptographic Keys, Cryptographic Key Components, and CSPs

CSP/Key	CSP/Key Type	Input	Output	Storage	Zeroization	Use
AES key	128, 192, 256-bit AES key	Internally generated via ANSI x9.31 PRNG	API call parameter	Plaintext in volatile memory	By power cycle or host reboot	Encryption, decryption
TDES key	168-bit Triple-DES key	Internally generated via ANSI x9.31 PRNG	API call parameter	Plaintext in volatile memory	By power cycle or host reboot	Encryption, decryption
HMAC key	HMAC key	Internally generated via ANSI x9.31 PRNG	API call parameter	Plaintext in volatile memory	By power cycle or host reboot	Message Authentication with SHS

CSP/Key	CSP/Key Type	Input	Output	Storage	Zeroization	Use
RSA private key	3072-bit RSA key	Input electronically in plaintext	Never exits the module	Plaintext in volatile memory	By power cycle or host reboot	Decryption
		Internally generated via ANSI x9.31 PRNG	API call parameter			Used by calling application
RSA public key	3072-bit RSA key	Input electronically in plaintext	Never exits the module	Plaintext in volatile memory	By power cycle or host reboot	Signature verification, encryption
		Internally generated via ANSI X9.31 PRNG	API call parameter			Used by calling application
DH public components	Public components of DH protocol	Internally generated	API call parameter	Plaintext in volatile memory	By power cycle or host reboot	Used by calling application
DH private component	Private exponent of DH protocol	Internally generated	Never exits the module	Plaintext in volatile memory	By power cycle or host reboot	Used by calling application
ANSI X9.31 PRNG seed	128-bit random value	Externally generated using an NDRNG[21] and input in plaintext	Never exits the module	Plaintext in volatile memory	By power cycle or host reboot	Generate random number
ANSI X9.31 PRNG seed key	128-bit AES key	Externally generated using an NDRNG and input in plaintext	Never exits the module	Plaintext in volatile memory	By power cycle or host reboot	Generate random number

2.7.1 Key Generation

The module uses an ANSI X9.31 Appendix A.2.4 PRNG implementation to generate cryptographic keys[22]. This PRNG is FIPS-Approved (as shown in Annex C to FIPS PUB 140-2). The module also supports the generation of the RSA and Diffie-Hellman (DH) public/private keys using the RSA key generation function specified in ANSI X9.31.

[21] NDRNG – Non-Deterministic Random Number Generator

[22] Caveat: The module generates cryptographic keys whose strengths are modified by available entropy (no assurance of the minimum strength of generated keys).

2.7.2 Key Entry and Output

The cryptographic module itself does not support key entry or key output. However, keys are passed to the module as parameters from the applications resident on the host platform via the exposed APIs. Keys that enter the module via an API call parameter are in plaintext. Similarly, keys and CSPs exit the module in plaintext via the APIs.

2.7.3 Key/CSP Storage and Zeroization

As a software module, the module does not provide for the persistent storage of keys and CSPs. Keys and CSPs stored in RAM can be zeroized by a power cycle or a host platform reboot. Additionally, symmetric, asymmetric, and HMAC keys are either provided by or delivered to the calling process, and are subsequently destroyed by the module at the completion of the API call. The ANSI X9.31 PRNG seed and seed key are initialized by the module (externally generated and entered into the module in plaintext) at power-up and remain stored in RAM until the module is uninitialized by a host platform reboot or power cycle.

The key zeroization techniques used for clearing volatile memory, once invoked, take effect immediately, and do not allow sufficient time to compromise any plaintext secret and private keys and CSPs stored by the module.

2.8 EMI/EMC

The McAfee Vulnerability Manager Cryptographic Module is a software module. Therefore, the only electromagnetic interference produced is that of the host platform on which the module resides and executes. FIPS 140-2 requires that the host platforms on which FIPS 140-2 testing is performed meet the Federal Communications Commission (FCC) EMI and EMC requirements for business use as defined in Subpart B, Class A of FCC 47 Code of Federal Regulations Part 15. The host system meets these FCC requirements and the FCC certificate can be provided by the manufacturer of the hardware.

2.9 Self-Tests

The McAfee Vulnerability Manager Cryptographic Module performs a set of self-tests upon power-up and conditionally during operation as required in FIPS 140-2.

2.9.1 Power-Up Self-Tests

The McAfee VMCM performs the following self-tests at power-up:

- Software integrity check using HMAC SHA-1
- Known Answer Tests (KATs) for:
    - AES
    - Triple-DES
    - SHA-1
    - HMAC SHA-1
    - HMAC SHA-512
    - RSA (sign/verify)
    - ANSI X9.31 RNG

(Note: SHA-512 is tested as part of HMAC SHA-512 KAT.)

2.9.2 Conditional Self-Tests

The module performs the following conditional self-tests:

- Continuous RNG test

- RSA Pairwise Consistency test

2.9.3 Self-Test Error Condition

If any self-test fails, the module will enter a critical error state, during which cryptographic functionality and all data output is inhibited. To clear the error state, the CO must reboot the host platform.

2.10 Mitigation of Other Attacks

This section is not applicable. The modules do not claim to mitigate any attacks beyond the FIPS 140-2 Level 1 requirements for this validation.

3 Secure Operation

The McAfee Vulnerability Manager Cryptographic Module meets Level 1 requirements for FIPS 140-2. The sections below describe how to place and keep the module in its FIPS-Approved mode of operation. Section 3.1 below provides guidance to the Crypto Officer for managing the module.

3.1 Secure Management

The following sections provide the necessary guidance to ensure that the module is running in its Approved mode of operation.

3.1.1 Initialization

When the module is installed and initialized, the module is considered to be running in its Approved mode of operation. This is achieved by calling a single initialization function *FIPS_mode_set()* with the parameter “1”, which is automatically invoked on the first call to *SSL_library_init().* Upon initialization of the module, the module requires no set-up and runs its suite of power-up self-tests, which includes a software integrity test that checks the integrity of the module using an HMAC SHA-1 digest. If the integrity check succeeds, then the module performs cryptographic algorithm self-tests. If the module passes all the self-tests, the function will set an internal global variable and then return a value of “1” to the calling application to indicate that the module is in a Approved mode of operation or “0” to indicate test failure.

3.1.2 Management

Since the Crypto Officer cannot directly interact with the module, no specific management activities are required to ensure that the module runs securely; the module only executes in an Approved mode of operation when operated according to this Security Policy. If any irregular activity is noticed or the module is consistently reporting errors, then McAfee, Inc. Customer Support should be contacted.

Power-up self-tests can be performed on demand by cycling the power on the host platform, by calling the function *FIPS_selftest(),* or by reinitializing the module by using the *FIPS_mode_set()* function via *SSL_library_init().*

3.1.3 Zeroization

The module does not persistently store any key or CSPs. All ephemeral keys used by the module are zeroized upon session termination. All keys can be zeroized by power cycling or rebooting the host system.

3.2 User Guidance

It is the responsibility of the calling application developer to ensure that only appropriate algorithms, key sizes, and key establishment techniques are applied. Users are responsible for using only the services that are listed in Table 5. Any use of the McAfee Vulnerability Manager Cryptographic Module with non-Approved cryptographic services or keys that provide less than 112 bits of encryption strength constitutes a departure from this Security Policy, and results in the module not being in its Approved mode of operation.

Although the User does not have any ability to modify the configuration of the module, they should notify the Crypto Officer if any irregular activity is noticed.

3.3 Non-Approved Mode of Operation

When installed, initialized, and configured according to the Crypto-Officer guidance in this Security Policy, the module does not support a non-Approved mode of operation.

4 Acronyms

Table 7 below defines the acronyms used in this document.

Table 7 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
ANSI	American National Standards Institute
API	Application Programming Interface
CAPI	Cryptography Application Programming Interface
CAST	Carlisle Adams and Stafford Tavares
CBC	Cipher Block Chaining
CD	Compact Disc
CMVP	Cryptographic Module Validation Program
CNG	CAPI: Next Generation
CO	Crypto Officer
CPU	Central Processing Unit
CSEC	Communications Security Establishment Canada
CSP	Critical Security Parameter
DES	Data Encryption Standard
DH	Diffie-Hellman
DVD	Digital Video Disc
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
FCC	Federal Communications Commission
FIPS	Federal Information Processing Standard
FISMA	Federal Information Security Management Act of 2002
GOST	Gosudarstvennyy Standart (translates from German to "State Standard")
GPC	General Purpose Computer
HIPAA	Health Insurance Portability and Accountability Act of 1996
HMAC	(Keyed-) Hash Message Authentication Code
IDEA	International Data Encryption Algorithm
IIS	Internet Information Services
KAS	Key Agreement Scheme
KAT	Known Answer Test
LED	Light Emitting Diode

Acronym	Definition
MAC	Message Authentication Code
MD	Message Digest
MDC	Modification Detection Code
N/A	Not Applicable
NDRNG	Non-Deterministic Random Number Generator
NIST	National Institute of Standards and Technology
OS	Operating System
PCI-DSS	Payment Card Industry – Data Security Standard
PKCS	Public-Key Cryptography Standards
PRNG	Pseudo-Random Number Generator
PSS	Probabilistic Signature Scheme
PUB	Publication
RAM	Random Access Memory
RC	Rivest Cipher
RNG	Random Number Generator
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SOX	Sarbanes-Oxley Act of 2002
SP	Special Publication
SQL	Structured Query Language
SSL	Secure Socket Layer
TDES	Triple Data Encryption Standard
USB	Universal Serial Bus
VMCM	Vulnerability Manager Cryptographic Module

Prepared by:
Corsec Security, Inc.

13135 Lee Jackson Memorial Highway, Suite 220
Fairfax, VA 22033
United States of America

Phone: +1 (703) 267-6050
Email: info@corsec.com
http://www.corsec.com