Protect what you value.

McAfee, Inc.

McAfee Endpoint Encryption for Files and Folders

FIPS 140-2 Non-Proprietary Security Policy

Level 1 Validation

Document revision 0.6, April 2009

McAfee, Inc.
3965 Freedom Circle
Santa Clara, CA 95054,
888.847.8766
www.mcafee.com

Table of Contents

Table of Figures

1 INTRODUCTION

1.1 Purpose

This is the non-proprietary FIPS 140-2 Security Policy for the McAfee Endpoint Encryption for Files and Folders cryptographic module, also referred to as “the module” within this document. This Security Policy details the secure operation of McAfee Endpoint Encryption for Files and Folders as required in Federal Information Processing Standards Publication 140-2 (FIPS 140-2) as published by the National Institute of Standards and Technology (NIST) of the United States Department of Commerce.

1.2 References

For more information on McAfee Endpoint Encryption please visit: http://www.mcafee.com/us/enterprise/products/data_loss_prevention/endpoint_encryption.html. For more information on NIST and the Cryptographic Module Validation Program (CMVP), please visit http://csrc.nist.gov/groups/STM/cmvp/index.html.

1.3 Document Organization

This Security Policy document is one part of the FIPS 140-2 Submission Package. This document outlines the functionality provided by the module and gives high-level details on the means by which the module satisfies FIPS 140-2 requirements. With the exception of this Non-Proprietary Security Policy, the FIPS 140-2 Submission documentation may be McAfee, Inc. proprietary or otherwise controlled and releasable only under appropriate non-disclosure agreements. For access to these documents, please contact McAfee, Inc.

2 McAfee Endpoint Encryption for Files and Folders

McAfee Endpoint Encryption for Files and Folders (SW Version 3.1.1.7), also referred to simply as “module”, is a Software Only Module which resides on a General Purpose Computer (see Figure 1). It provides a security system for PCs that encrypts assets in the form of files and folders on a user’s PC according to a centrally managed security policy. McAfee Endpoint Encryption for Files and Folders provides “Persistent Encryption Technology (PET)”, which ensures that the protection travels with the document and allows sharing of encrypted data.

The cryptographic boundary of the module is the case of the device on which it is installed. See Figure 1. The module is a software module running on a standard General Purpose Computing (GPC) device. The processor of the GPC device executes all software. All software components of the module are persistently stored within the device and, while executing, are stored in the device local RAM.

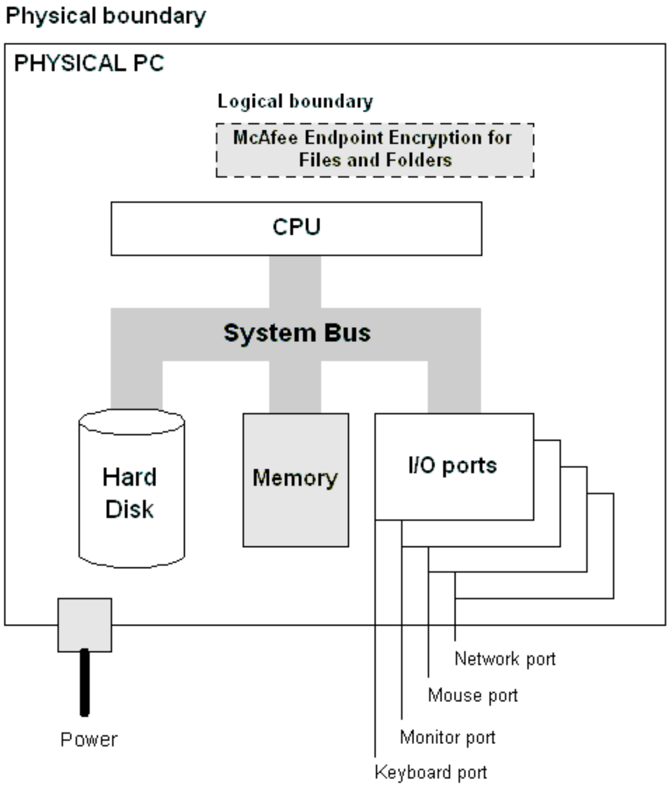

Figure 1: Block Diagram of the cryptographic boundary

The cryptographic module meets the overall requirements applicable to Level 1 security of FIPS 140-2.

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

Figure 2: Security Level specification per individual areas of FIPS 140-2

The McAfee Endpoint Encryption for Files and Folders encrypts files and folders according to policies set by remote McAfee Endpoint Encryption Administration Servers (these servers are outside the physical and logical boundaries of the module). McAfee Endpoint Encryption for Files and Folders acts like a filter between applications creating or editing files and using storage media. Whenever a file is written to supported storage media, the McAfee Endpoint Encryption for Files and Folders encryption engine checks the encryption policy of the module and handles the file or folder accordingly. If the file or folder is configured by policy to be encrypted, then the encryption engine executes assigned encryption policies and encrypts the folder or file. If the file or folder is not included in the encryption policy then the file or folder bypasses the encryption functionality and is eventually written to the local storage media in plaintext. When an application later reads the folder or file, the encryption engine automatically decrypts the file when it is read into memory.

The encryption/decryption process happens automatically and is fully transparent. The user should not notice any difference between working with encrypted and plaintext files.

When a file is encrypted, it is encrypted at its original location on the disk. Hence, no copies or other special files are created when encrypting a file. The original file remains encrypted at all times, only the parts read into the memory are decrypted when an application reads the file.

2.1 McAfee Endpoint Encryption for Files and Folders

McAfee Endpoint Encryption for Files and Folders consists of a Windows Mini Filter driver, other Windows drivers, a system tray application and a set of Windows Dynamic Link Libraries (DLLs). These components comprise the validated module. McAfee Endpoint Encryption installs these files on the user's hard drive, and when the user logs onto their PC they are presented with the McAfee Endpoint Encryption for Files and Folders logon screen.

2.2 Module Interfaces

McAfee Endpoint Encryption for Files and Folders is classified as a multi-chip standalone module for FIPS 140-2 purposes. The module's physical boundary is that of the Windows Personal Computer (PC) on which it is installed. The Windows PC shall be running a supported operating system (OS) and in addition support all standard interfaces, including a computer keyboard, mouse, screen, LAN ports, floppy drive, CD-ROM drive, speaker, disk drive, microphone inputs, serial ports, parallel ports, and power plug.

McAfee Endpoint Encryption for Files and Folders provides a logical interface via an Application Programming Interface (API) and a Graphical User Interface (GUI). This logical interface exposes services (described in section 2.4) that the User and operating system may utilize directly.

The logical interfaces provided by McAfee Endpoint Encryption for Files and Folders are mapped onto the FIPS 140-2 logical interfaces: data input, data output, control input, and status output as follows:

- Data Input – Input to all driver functions, Software Interface (SWI), GUI
- Data Output – Output from all driver functions, GUI, Software Interface (SWI)
- Control Input – Input from TCP/IP interface, IPC interface, GUI, Software Interface (SWI)
- Status Output – Return codes from driver functions, GUI, Software Interface (SWI)

2.3 Operational Environment

The McAfee Endpoint Encryption for Files and Folders module has been tested on and found to be conformant with the requirements of FIPS 140-2 overall Level 1 on the following GPC operating systems:

- Microsoft Windows Vista 64
- Microsoft Windows XP
- Microsoft Windows Vista 32

The module is also supported on Microsoft Windows 2000.

The cryptographic module runs in its own operating system threads. This provides it with protection from all other processes, preventing access to all keys, intermediate key generation values, and other CSPs.

The task scheduler and architecture of the operating system maintain the integrity of the cryptographic module.

The module supports only one single user and only one operator can have access to the device that contains the module at a time. For the purposes of FIPS 140-2, each of the Windows operating systems listed above must be configured as a single user operating system.

2.4 Roles and Services

McAfee Endpoint Encryption for Files and Folders implements both a Crypto Officer role and a User role. Roles are assumed implicitly upon accessing the associated services as depicted below. Figure 4 summarizes the services available to each role.

Role	Description
Crypto Officer	Administrator of the module having full configuration and key management privileges.
User	General User of the module

Figure 3: Roles

2.5 Access to Services

The following table, Figure 4, lists the authorized services linked to each of the Roles offered by the module.

Role	Authorized Services	Description
User	Synchronization	Establishes a secure network connection between the module and the remote Admin Server for the purpose of configuring the module.
	Encryption/Decryption	Encryption/Decryption of data written to the hard drive.
	Self-test Functions	Performs all FIPS 140-2 defined self tests.
	Recovery request	If the User is denied access to the module then the recovery request can be used to re-allow access. (Note: Successful utilization of this service requires Crypto Officer assistance.)
	Uninstall	Uninstalls the module from the host platform and zeroizes the Server Public Key
	Power Cycle	Power Cycles the host device and zeroizes CSPs in RAM
	Zeroize	Actively overwrites the Device Encryption Key.
Crypto Officer	Synchronization	Establishes a secure network connection between the module and the remote Admin Server for the purpose of configuring the module.
	Configuration	Configuration of the module.
	View audit	View audit log information.
	Clear audit	Deletes the audit log.
	File Updates	This service is used to update or add-on module functionality as opposed to performing a full software update.
	Manage McAfee Endpoint Encryption for Files and Folders (Cryptographic & Key Management Functions)	Manage McAfee Endpoint Encryption for Files and Folders, including Cryptographic & Key Management Functions and the encryption policy. The encryption policy specifies which files and folders are encrypted. Files and folders which are not configured to be encrypted are stored in plaintext, and therefore bypass the encryption mechanism.
	Software Updates	Loads new module software.
	Recovery	Will allow a User to access a module to which they are currently disallowed access.
	Set Attributes	Set module Attributes
	Change Attributes	Change module Attributes

Figure 4: Services Authorized for Roles

2.6 Physical Security

McAfee Endpoint Encryption for Files and Folders is a software only cryptographic module and therefore the physical security requirements of FIPS 140-2 do not apply.

2.7 Cryptographic Key Management

The following tables list all Critical Security Parameters (CSPs) and public keys used within the McAfee Endpoint Encryption module. Currently, AES-256 is the only Approved encryption algorithm in McAfee Endpoint Encryption for Files and Folders product and all encryption keys are AES-256 keys. The server public key is a DSA key.

Key type	Purpose
Device Encryption Key	To encrypt local storage, application databases, and external storage.
Session Key	Key used to encrypt traffic between device and remote server
Diffie-Hellman Shared Secret	Shared secret generated by the Diffie-Hellman Key exchange.
Diffie-Hellman Private Key	Private Diffie-Hellman component used during Session Key agreement.
DRNG Seed Key	Seed values and seed key used as input into the FIPS 186-2 DRNG.

Figure 5: CSPs used by McAfee Endpoint Encryption for Files and Folders

Key type	Purpose
Server Public Key	DSA Key used to authenticate software during power-up self tests and software updates. It is also used to authenticate the Diffie-Hellman Server Public Key entry during Synchronization.
Diffie-Hellman Server Public Key	The Server Public Diffie-Hellman component used during Session Key agreement.
Diffie-Hellman Client Public Key	The Client Public Diffie-Hellman component generated internally by the module and used during Session Key agreement.

Figure 6: Public Keys used by McAfee Endpoint Encryption for Files and Folders

2.7.1 Key generation

McAfee Endpoint Encryption for Files and Folders generates symmetric key material and CSPs (and the Diffie-Hellman public/private key components used in session CSP establishment) using a FIPS 186-2 Appendix 3.1 compliant deterministic random number generator. The only symmetric keys/CSPs generated in this way are the Device Encryption Key and the Session Key.

McAfee Endpoint Encryption for Files and Folders also supports Session Key Derivation included as part of the Diffie-Hellman Key exchange, which is allowed for FIPS 140-2.

2.7.2 Key entry and output

The module supports the following key entry:

- Entry of the Diffie-Hellman Server Public Key signed with the Server Private Key

The module supports the following key output:

- Plaintext output of the Diffie-Hellman Client Public Key
- Encrypted output of the Device Encryption Key

2.7.3 Key storage

Key material is stored in the McAfee Endpoint Encryption datastore in local GPC storage.

2.7.4 Zeroization of key material

All key material managed by the McAfee Endpoint Encryption for Files and Folders has the ability to be zeroized.

2.7.5 Access to key material

The following matrices (Figures 7 and 8) show the access that an operator has to specific keys or other critical security parameters when performing each of the services relevant to his/her role.

	Key							
Service	**DK**	**SPK**	**DRNGSK**	**DHK**	**DHSPK**	**DHCPK**	**DHSS**	**SK**
Synchronization	R, O	R, W	R, W	W	W, E	W, O	W, R	R, W
Encryption/Decryption	R							
Self-test Functions								
Recovery Request								
Uninstall		Z						
Power Cycle	Z		Z	Z	Z	Z	Z	Z
Zeroize	Z							

Figure 7: User Role

	Key							
Service	**DK**	**SPK**	**DRNGSK**	**DHK**	**DHSPK**	**DHCPK**	**DHSS**	**SK**
Synchronization	R, O	R	R, W	W	W, E	W, O	W, R	R, W
View audit		R						R
Clear audit		R						R
Configuration	R	R						R
File Updates		R						R
Manage McAfee Endpoint Encryption for Files and Folders (Cryptographic & Key Management Functions)	R	R						R
Software Updates		R, E, W						R
Recovery	R							R
Set Attributes		R						R
Change Attributes		R						R

Figure 8: Crypto-Officer Role

Access rights	
Blank	Not Applicable
W	Write access
R	Read Access
E	Key Entry
O	Key Output
Z	Zeroize Access

Keys	
DK	Device Encryption Key
SPK	Server Public Key
DRNGSK	DRNG Seed Key
DHK	Diffie-Hellman Private Key
DHSS	Diffie-Hellman Shared Secret
DHSPK	Diffie-Hellman Server Public Key
DHCPK	Diffie-Hellman Client Public Key
SK	Session Key

Note: If a service requires read or write access, it is the service as realized by module processes that requires access to the keys or CSPs. The operator (either User or Crypto Officer) does not have access to the CSPs themselves. The operator may change keys or use keys, but in all cases has no plaintext access to key material or CSPs.

2.8 Cryptographic Algorithms

McAfee Endpoint Encryption for Files and Folders supports the following algorithms:

- FIPS-approved algorithms
  - AES-256
  - DSA
  - SHA-1
  - FIPS 186 Appendix 3.1 DRNG.
- Non FIPS-approved algorithms allowed for use in FIPS mode:
  - Diffie-Hellman (key establishment methodology provides 80 bits of security)
  - NDRNG (Used to seed the FIPS approved DRNG)
- Non FIPS-approved algorithms only used in non-FIPS mode:
  - RC5 (12 rounds)
  - RC5 (18 rounds)
  - AES 256 (non-compliant)

2.9 Self-Tests

McAfee Endpoint Encryption for Files and Folders implements both power-up and conditional self tests as required by FIPS 140-2. The following two sections outline the tests that are performed.

2.9.1 Power-up self-tests

The following table, Figure 9, lists the power-up self-tests performed by the module:

SHA-1 known answer test
DSA known answer test (DSA Signature Verification)
AES-256 known answer test (Encrypt/Decrypt)
Software integrity test (DSA Signature verification)
Deterministic Random Number Generator Known Answer Test

Figure 9: Power-up Self-tests

Each of these tests is executed when the computer is turned on and the module first executes. If any of these tests fail, the module will not load. The module must be reset to re-execute these tests.

2.9.2 Conditional self-tests

There are a number of conditional tests that are run by the module. A continuous random number generator test is run every time the module requests a random number from either the FIPS Approved 186-2 DRNG or the NDRNG. Failure of this test may result in keys not being generated and an appropriate error message will be given. A test is also done when a software update or file update occurs. All files are digitally signed and this signature is checked prior to any update of the software.

The final self-test performed by the module is a bypass test, which ensures that the bypass mechanism is functioning properly. The bypass test ensures that the encryption policy has not been altered since the last modification by comparing a stored hash of the encryption policy to the existing policy. The second portion of the bypass test forces a known answer test of the encryption/decryption algorithm to ensure that it is correctly functioning when the encryption policy is modified.

2.10 Design Assurance

McAfee, Inc. employ industry standard best practices in the design, development, production and maintenance of the McAfee Endpoint Encryption product, including the FIPS 140-2 module.

This includes the use of an industry standard configuration management system that is operated in accordance with the requirements of FIPS 140-2, such that each configuration item that forms part of the module is stored with a label corresponding to the version of the module and that the module and all of its associated documentation can be regenerated from the configuration management system with reference to the relevant version number.

Design documentation for the module is maintained to provide clear and consistent information within the document hierarchy to enable transparent traceability between corresponding areas throughout the document hierarchy, for instance, between elements of this Cryptographic Module Security Policy (CMSP) and the design documentation.

Guidance appropriate to an operator’s Role is provided with the module and provides all of the necessary assistance to enable the secure operation of the module by an operator, including the Approved security functions of the module.

3 FIPS Mode

The following procedures must be followed to operate McAfee Endpoint Encryption for Files and Folders cryptographic module in a FIPS Approved mode:

1. McAfee Endpoint Encryption for Files and Folders must be configured to use only FIPS Approved algorithms. The validated version of McAfee Endpoint Encryption for Files and Folders presents AES-256 as the only option for the encryption algorithm. The AES-256 encryption algorithm is validated for use in FIPS 140-2 implementations.
2. The module software must be operating in “FIPS” mode. This is done by setting the FIPS registry key value from 0 (disabled) to 1 (enabled). The first step is to create a FIPS registry script (see Appendix A for details). Once the file is created, right click on the newly created .reg file and select merge from the drop down menu.
3. To verify that the registry has been updated properly the user must install a registry editor and navigate to the following paths and verify that “FipsMode is set to 1”:
    a. Windows 2000 and XP - HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier
    b. HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier
4. The PC used to run McAfee Endpoint Encryption for Files and Folders must be built using production grade components and configured in a single operator mode. To do this, the following operating system services must be disabled:
    - Fast user switching
    - Terminal services
    - Remote registry service
    - Secondary logon service
    - Telnet service
    - Remote desktop and Remote assistance services

.

4 Appendix A – FIPS mode registry script

The following needs to be saved to a text file with the extension ".reg" and then merged into the registry as a requirement for installing the module in a FIPS-compliant mode of operation:

4.1 Windows 2000/XP

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier]
"FipsMode"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\1]
"Path"="c:\\program files\\safeboot content encryption\\SbCeProvider.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\2]
"Path"="c:\\program files\\safeboot content encryption\\SbCeNp.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\3]
"Path"="c:\\program files\\safeboot content encryption\\SbCeDe5Auth.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\4]
"Path"="c:\\program files\\safeboot content encryption\\SbCeSetup.exe

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\5]
"Path"="c:\\program files\\safeboot content encryption\\SbCeObj.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\6]
"Path"="c:\\program files\\safeboot content encryption\\SbKeysObj.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\7]
"Path"="c:\\program files\\safeboot content encryption\\SbCeMarshal.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\8]
"Path"="c:\\program files\\safeboot content encryption\\SbCmaCe.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\9]
"Path"="c:\\program files\\safeboot content encryption\\SbAlgs\\SbAlg00.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\10]
"Path"="c:\\program files\\safeboot content encryption\\SbAlgs\\SbAlg01.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\11]
"Path"="c:\\program files\\safeboot content encryption\\SbAlgs\\SbAlg11.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\12]
"Path"="c:\\program files\\safeboot content encryption\\SbAlgs\\SbAlg12.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\13]
"Path"="c:\\windows\\system32\\drivers\\Sbalg00.sys"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\14]
"Path"="c:\\windows\\system32\\drivers\\Sbalg01.sys"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\15]
"Path"="c:\\windows\\system32\\drivers\\Sbalg11.sys"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\16]
"Path"="c:\\windows\\system32\\drivers\\Sbalg12.sys"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\17]
"Path"="c:\\program files\\safeboot content encryption\\SbComms.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\18]
"Path"="c:\\program files\\safeboot content encryption\\SbTokens\\SbTokenPwd.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\19]
"Path"="c:\\program files\\safeboot content encryption\\SbCeCore.exe"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\20]
"Path"="c:\\program files\\safeboot content encryption\\SbCeCoreService.exe"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\21]
"Path"="c:\\program files\\safeboot content encryption\\SbCeDesktopIntegration.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\22]
"Path"="c:\\windows\\system32\\drivers\\SbCe.sys"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\23]
"Path"="c:\\windows\\system32\\drivers\\SbCeCd.sys"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\24]
"Path"="c:\\program files\\safeboot content encryption\\SbCeDriverCom.dll

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\25]
"Path"="c:\\program files\\safeboot content encryption\\SbCeLocalProvider.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\26]
"Path"="c:\\program files\\safeboot content encryption\\SbCePolicyEnforcer.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\27]
"Path"="c:\\program files\\safeboot content encryption\\SbCeProviderManager.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\28]
"Path"="c:\\program files\\safeboot content encryption\\SbCeTray.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\29]
"Path"="c:\\program files\\safeboot content encryption\\resource.dll"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\30]
"Path"="c:\\program files\\safeboot content encryption\\SbCeSelfExtractorStub.exe"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\31]
"Path"="c:\\program files\\safeboot content encryption\\SbCeShell.exe"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\32]
"Path"="c:\\program files\\safeboot content encryption\\SbCeShell.com"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\33]
"Path"="c:\\program files\\safeboot content encryption\\SbCeProxy.exe"

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\34]
"Path"="c:\\program files\\safeboot content encryption\\SbCePostInstall.dll"
```

4.2 Windows Vista

```
REGEDIT4

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier]
"FipsMode"=dword:00000001

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\1]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeProvider.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\2]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeNp.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\3]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeDe5Auth.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\4]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeSetup.exe

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\5]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeObj.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\6]
“Path”=“c:\\program files\\safeboot content encryption\\SbKeysObj.dll”
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\7]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeMarshal.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\8]
“Path”=“c:\\program files\\safeboot content encryption\\SbCmaCe.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\9]
“Path”=“c:\\program files\\safeboot content encryption\\SbAlgs\\SbAlg00.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\10]
“Path”=“c:\\program files\\safeboot content encryption\\SbAlgs\\SbAlg01.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\11]
“Path”=“c:\\program files\\safeboot content encryption\\SbAlgs\\SbAlg11.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\12]
“Path”=“c:\\program files\\safeboot content encryption\\SbAlgs\\SbAlg12.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\13]
“Path”=“c:\\windows\\system32\\drivers\\Sbalg00.sys”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\14]
“Path”=“c:\\windows\\system32\\drivers\\Sbalg01.sys”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\15]
“Path”=“c:\\windows\\system32\\drivers\\Sbalg11.sys”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\16]
“Path”=“c:\\windows\\system32\\drivers\\Sbalg12.sys”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\17]
“Path”=“c:\\program files\\safeboot content encryption\\SbComms.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\18]
“Path”=“c:\\program files\\safeboot content encryption\\SbTokens\\SbTokenPwd.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\19]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeCore.exe”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\20]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeCoreService.exe”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\21]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeDesktopIntegration.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\22]
“Path”=“c:\\windows\\system32\\drivers\\SbCe.sys”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\23]
“Path”=“c:\\windows\\system32\\drivers\\SbCeCd.sys”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\24]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeDriverCom.dll

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\25]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeLocalProvider.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\26]
“Path”=“c:\\program files\\safeboot content encryption\\SbCePolicyEnforcer.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\27]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeProviderManager.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\28]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeTray.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\29]
“Path”=“c:\\program files\\safeboot content encryption\\resource.dll”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\30]
```

```
“Path”=“c:\\program files\\safeboot content encryption\\SbCeSelfExtractorStub.exe”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\31]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeShell.exe”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\32]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeShell.com”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\33]
“Path”=“c:\\program files\\safeboot content encryption\\SbCeProxy.exe”

[HKEY_LOCAL_MACHINE\SOFTWARE\SafeBoot International\SafeBoot Content Encryption\Verifier\34]
“Path”=“c:\\program files\\safeboot content encryption\\SbCePostInstall.dll”
```

McAfee, Inc.
3965 Freedom Circle
Santa Clara, CA 95054,
888.847.8766
www.mcafee.com