Boot Manager (bootmgr) Security Policy

For FIPS 140-2 Validation

v 3.3
12/27/07

1 Introduction

The Windows Boot Manager (BOOTMGR, version 6.0.6000.16386) is the system boot manager, called by the bootstrapping code that resides in the boot sector. BOOTMGR is responsible for loading and verifying the integrity of the Windows OS Loader, Winload.exe.

1.1 Cryptographic Boundary for BOOTMGR

The Windows Vista Boot Manager consists of a single executable. The cryptographic boundary for BOOTMGR is defined as the enclosure of the computer system, on which BOOTMGR is to be executed. The physical configuration of BOOTMGR, as defined in FIPS-140-2, is multi-chip standalone.

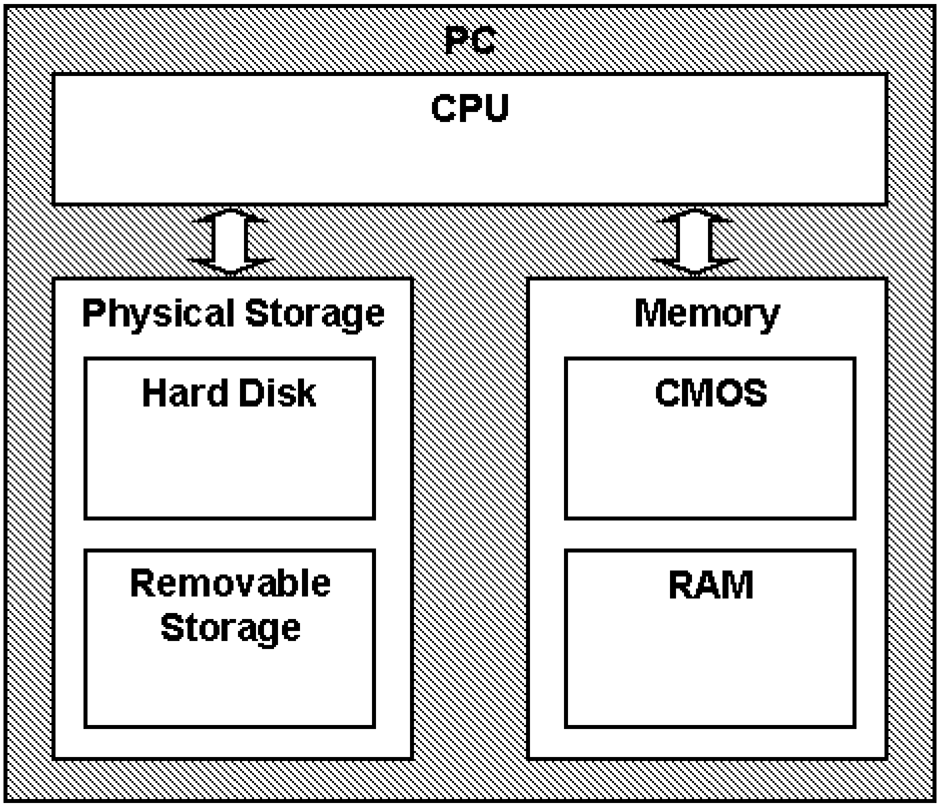
2 Security Policy

2.1 Boot Manager Security Policy

BOOTMGR operates under several rules that encapsulate its security policy.

- BOOTMGR is supported on Windows Vista.
- Windows Vista is an operating system supporting a "single user" mode where there is only one interactive user during a logon session.
- BOOTMGR is only in its Approved mode of operation when Windows is booted normally, meaning Debug mode is disabled and Driver Signing enforcement is enabled.

The following diagram illustrates the master components of the BOOTMGR module



- BOOTMGR's primary service is to load the Windows Vista operating system loader (Winload), after it determines Winload's integrity using its cryptographic algorithm implementations using the FIPS 140-2 approved algorithms mentioned below. After the verified Winload binary image file is loaded, BOOTMGR passes the execution control to Winload and no longer executes until the next reboot. The Crypto office and User have access to the service BOOTMGR supports.
- If the integrity of Winload is not verified, BOOTMGR does not transfer the execution from itself to Winload.
- BOOTMGR is also responsible for loading the WinResume component when booting from hibernation.
- Boot Manager also loads any boot utilities (such as memtest) as part of its non-Approved services.
- Boot Manager implements a self-integrity check using an RSA digital signature during its initialization process. Boot Manager will not complete its initialization if the signature is invalid.
- BOOTMGR implements the following FIPS-140-2 Approved algorithms.
    - RSA PKCS#1 (v1.5) digital signature verification (Cert. #255)
    - SHS (Cert. #618)
    - AES (Cert. #424)
    - HMAC-SHA-256 (Cert. #298)[1]

Cryptographic bypass is not supported by BOOTMGR.

BOOTMGR was tested using the following machine configurations:

X86	Microsoft Windows Vista Ultimate Edition (x86 version) – Dell SC420 (Intel Pentium 2.53GHz)
AMD64	Microsoft Windows Vista Ultimate Edition (x64 version) – Dell SC430 (Intel Pentium D 2.8GHz)

3 BOOTMGR Ports and Interfaces

3.1 Control Input Interface

The BOOTMGR Data Input Interface is the set of internal functions responsible for intercepting control input. These functions are:

- BlBdInitialize – Determines if a boot debugger is attached.
- BlInitializeLibrary – Intercepts parameters from diskboot.asm.
- BmOpenBootIni – Parses information contained in boot.ini.
- BmGetOptionList – Loads application options from the Boot Configuration Data store.
- BlXmiRead – Reads the operator selection from the Boot Manager UI.

3.2 Status Output Interface

The Status Output Interface is the BlXmiWrite function that is responsible for displaying the integrity verification errors to the screen. The Status Output Interface is also defined as the BlLogData responsible for writing status to the bootlog.

3.3 Data Output Interface

The Data Output Interface is made up of the following functions:
Archx86TransferTo32BitApplicationAsm, Archx86TransferTo64BitApplicationAsm,

[1] HMAC-SHA-256 is only used to self-test SHA-256. Please see section 6 for details.

Archpx64TransferTo64BitApplicationAsm. These functions are responsible for transferring the execution from Boot Manager to the initial execution point of Winload. Data exits the module in the form of the initial instruction address of Winload.

The Data Output Interface is also represented by the ImgpInitializeBootApplicationParameters function. This function is responsible for passing application parameters from Boot Manager to Winload.

3.4 Data Input Interface

The Data Input Interface is represented by the BlFileReadEx function and the BlDeviceRead function. BlFileReadEx is responsible for reading the binary data of unverified components from the computer hard drive. BlDeviceRead is used for reading data directly from devices.

Additionally, the GPC's USB port also forms a part of the Data Input interface. This interface is used to enter the ExK key used by the BitLocker™ Drive Encryption application shipped with the Enterprise and Ultimate Editions of Windows Vista.

4 Specification of Roles

BOOTMGR supports both User and Cryptographic Officer roles (as defined in FIPS-140-2). Both roles have access to all services implemented in BOOTMGR. The module does not implement any authentication services. Therefore, roles are assumed implicitly by booting the Windows Vista operating system.

4.1 Maintenance Roles

Maintenance roles are not supported by BOOTMGR.

4.2 Multiple Concurrent Interactive Operators

There is only one interactive operator during a boot session. Multiple concurrent interactive operators sharing a logon session are not supported.

5 Cryptographic Key Management

BOOTMGR does not store any secret or private cryptographic keys across power-cycles. However, it does use three AES keys in support of the BitLocker feature. These keys are:

- External Key (ExK) – 256-bit AES key stored outside the cryptographic boundary (e.g. – USB memory device). This key is entered into the module via the USB port and used to decrypt the VMK.
- Volume Master Key (VMK) – 256-bit AES key used to decrypt the FVEK and winload.exe.
- Full Volume Encryption Key (FVEK) – 128 or 256-bit AES key used to decrypt data on disk sectors.

Note that both the VMK and FVEK are stored in encrypted form across power cycles and thus not subject to the zeroization requirements. The ExK, is stored only in memory and is zeroized by rebooting the OS.

BOOTMGR also uses public keys stored on the computer hard disk to verify digital signatures using its implementation of RSA PKCS#1 (v1.5) verify. These public keys are available to both roles. Zeroization is performed by deleting the bootmgr module.

All the keys (mentioned above) are accessed only by the BOOTMGR service that loads the Windows Vista operating system loader (Winload). This service only has execute access to the keys mentioned above.

6 BOOTMGR Self Tests

BOOTMGR performs the following power-on (start up) self-tests.

- SHS (SHA-1) Known Answer Tests. As allowed by FIPS 140-2 IG 9.2, SHA-256 is tested via HMAC-SHA-256 Known Answer Test
- RSA PKCS#1 (v1.5) verify with public key
- AES Known Answer Tests

7 Additional details

For the latest information on Windows Vista, check out the Microsoft web site at http://www.microsoft.com.