Windows 7 Boot Manager Security Policy

For FIPS 140-2 Validation

v 1.1
4/26/10

1 Introduction

The Windows Boot Manager (BOOTMGR) – version 6.1.7600.16835 – is the system boot manager, called by the bootstrapping code that resides in the boot sector. BOOTMGR is responsible for capturing the credentials required to unlock the OS volume and for loading and verifying the integrity of the Windows OS Loader, Winload.exe, and Windows OS Resume, winresume.exe.

In Windows 7 the credentials supported for unlocking the OS volumes are:

- A startup key (stored on a USB flash drive; also known as an External key)
- A recovery key (stored on a USB flash drive; also known as an External key)
- An alpha-numeric PIN (for TPM+PIN or TPM+PIN+SK scenarios)

1.1 Cryptographic Boundary for BOOTMGR

The Windows 7 Boot Manager consists of a single executable and the Certificate Directory containing the root public key certificate issued by Microsoft. 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.

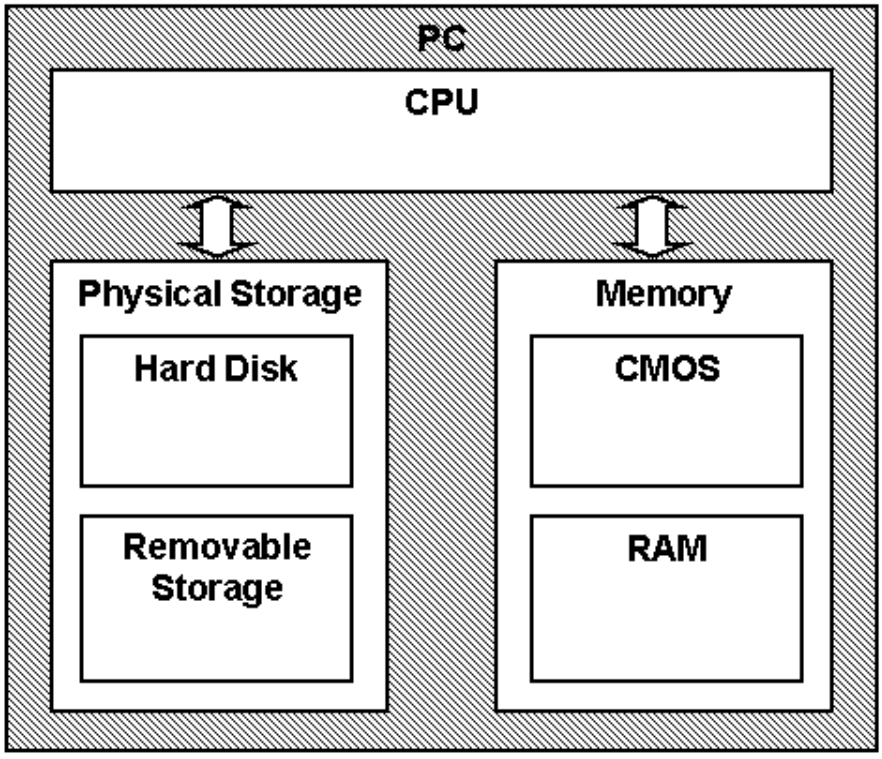
2 Security Policy

2.1 Boot Manager Security Policy

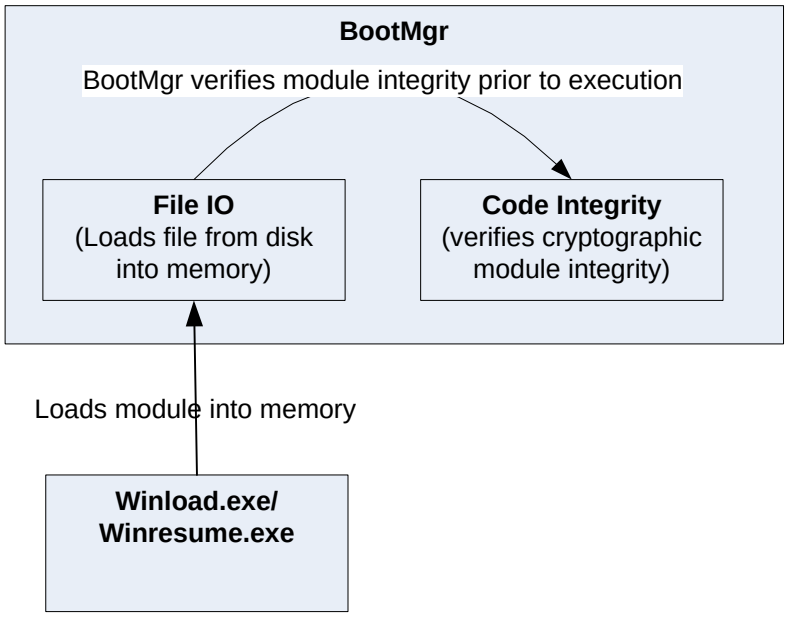
BOOTMGR operates under several rules that encapsulate its security policy.

- BOOTMGR is validated on Microsoft Windows 7 Ultimate Edition running on both x86 and x64 versions.
- Windows 7 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



The following diagram illustrates BootMgr module interaction with cryptographic module:



- BOOTMGR's only service is to load the Windows 7 operating system loader (Winload) or the Windows 7 OS Resume (winresume.exe), after it determines component's integrity using its cryptographic algorithm implementations using the FIPS 140-2 approved algorithms mentioned below. After the verified binary image file is loaded, BOOTMGR passes the execution control to it and no longer executes until the next reboot The Crypto office and User have access to the service BOOTMGR supports.
- If the integrity of components being loaded is not verified, BOOTMGR does not transfer the execution from itself.
- The module provides a power-up self-tests services that is automatically executed when the module is loaded into memory, as well as, a show status service, that is automatically executed by the module to provide the status response of the module either via output to the GPC monitor or to log files.
- 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. #557)
    - SHS (Cert. #1081)
    - AES (Certs. #1168 and 1177)
    - HMAC[1] (Cert. #675)
- BOOTMGR implements the following non-approved algorithms:
    - MD5 – used for certificate chain authentication

Cryptographic bypass is not supported by BOOTMGR.

3 BOOTMGR Ports and Interfaces

3.1 Control Input Interface

[1] HMAC is only used as an extraneous SHA-256 self-test.

The BOOTMGR Control Input Interface is the set of internal functions responsible for reading control input. These input signals are read from various system locations and are not directly provides by the operator. Examples of the internal function calls include:

- BlBdDebuggerEnabled – Reads the system flag to determines if the boot debugger is enabled.
- BlXmiRead – Reads the operator selection from the Boot Selection menu.
- BlGetBootOptionBoolean – Reads control input from a protected area of the Boot Configuration Data registry.

The GPC's keyboard can also be used as control input when it is necessary for an operator to provide a response to a prompt for input or in response to an error indicator.

3.2 Status Output Interface

The Status Output Interface is the BlStatusPrint function that is responsible for displaying the integrity verification errors to the screen. The Status Output Interface is also defined as the BsdpWriteAtLogOffset responsible for writing the name of the corrupt driver to the bootlog.

3.3 Data Output Interface

The Data Output Interface includes the following functions: Archx86TransferTo32BitApplicationAsm, Archx86TransferTo64BitApplicationAsm, Archpx64TransferTo64BitApplicationAsm. These functions are responsible for transferring the execution from Boot Manager to the initial execution point of Winload or Winresume. Data exits the module in the form of the initial instruction address of Winload or Winresume.

3.4 Data Input Interface

The Data Input Interface includes the BlFileReadEx function. BlFileReadEx is responsible for reading the binary data of unverified components from the computer hard drive.

Additionally, the GPC's USB port also forms a part of the Data Input interface. This interface is used to enter the Startup or Recovery Key key used by the BitLocker™ Drive Encryption application shipped with the Enterprise and Ultimate Editions of Windows 7. The GPC's keyboard can also serve as a Data Input Interface when the method to protect the VMK value relies on an operator supplied PIN.

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. Therefore, roles are assumed implicitly by booting the Windows 7 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 used 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 is the only method used to decrypt the VMK that results in the VMK being considered encrypted, as other methods rely upon non-approved key derivation methods.
- Intermediary Key (IK) – 256-bit AES key value used to decrypt the VMK. The value is not stored long-term and is derived based using a method defined by the system configuration. When the VMK is encrypted with this value, it is considered to be stored in plaintext.
- Volume Master Key (VMK) – 256-bit AES key used to decrypt the FVEK.
- Full Volume Encryption Key (FVEK) – 256-bit AES key used to encrypt data on disk sectors.

The VMK is always stored in encrypted form; however, based on system configuration the method used to perform the encryption may use a non-approved key derivation method meaning that the VMK is considered plaintext when stored. When encrypted with the ExK (identified above), the VMK is considered to be in an encrypted form, as no key derivation methods are used for the encrypting key. The VMK value can be zeroized by formatting the drive volume on which the key is stored.

Note that the FVEK is stored in encrypted form across power cycles and thus not subject to the zeroization requirements but can be zeroized in the same manner as the VMK. 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.

6 BOOTMGR Self Tests

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

- SHA (SHA-1/SHA-256/SHA-512) Known Answer Tests
- RSA Known Answer Tests
- Software Integrity Test – RSA PKCS#1 (v1.5) verify with public key
- AES Known Answer Tests

7 Additional details

For the latest information on Windows 7, check out the Microsoft web site at http://www.microsoft.com.