INTERNATIONAL STANDARD

ISO/IEC 19678

First edition
2015-05-01

Information Technology — BIOS Protection Guidelines

Technologies de l'information — Lignes directrices de protection BIOS

Reference number
ISO/IEC 19678:2015(E)

ISO copyright office
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Note: ITTF will provide the document number needed below

ISO/IEC 19678 was prepared by the U.S. National Institute of Standards and Technology from NIST SP 800-147, BIOS Protection Guidelines. NIST SP 800-147 was reformatted in accordance with ISO/IEC Directives, Part 2, while maintaining the technical content of the NIST publication (available at http://csrc.nist.gov/publications/nistpubs/800-147/NIST-SP800-147-April2011.pdf). The resulting standard was adopted under a special “fast-track procedure”, by Joint Technical Committee ISO/IEC JTC 1, *Information technology,* in parallel with its approval by the national bodies of ISO and IEC.

Introduction

Modern computers rely on fundamental system firmware, commonly known as the system Basic Input/Output System (BIOS), to facilitate the hardware initialization process and transition control to the operating system. The BIOS is typically developed by both original equipment manufacturers (OEMs) and independent BIOS vendors, and is distributed to end-users by motherboard or computer manufacturers. Manufacturers frequently update system firmware to fix bugs, patch vulnerabilities, and support new hardware. This International Standard provides security requirements and guidance for preventing the unauthorized modification of BIOS firmware on PC client systems.

Unauthorized modification of BIOS firmware by malicious software constitutes a significant threat because of the BIOS's unique and privileged position within the PC architecture. A malicious BIOS modification could be part of a sophisticated, targeted attack on an organization—either a permanent denial of service (if the BIOS is corrupted) or a persistent malware presence (if the BIOS is implanted with malware). The move from conventional BIOS implementations to implementations based on the Unified Extensible Firmware Interface (UEFI) may make it easier for malware to target the BIOS in a widespread fashion, as these BIOS implementations are based on a common specification.

This International Standard focuses on current and future x86 and x64 desktop and laptop systems, although the controls and procedures could potentially apply to any system design. Likewise, although the guide is oriented toward enterprise-class platforms, the necessary technologies are expected to migrate to consumer-grade systems over time. The security requirements do not attempt to prevent installation of unauthentic BIOSs through the supply chain, by physical replacement of the BIOS chip, or through secure local update procedures.

The intended audience for this International Standard includes BIOS and platform vendors, and information system security professionals who are responsible for managing the endpoint platforms' security, secure boot processes, and hardware security modules. The material may also be of use when developing enterprise-wide procurement strategies and deployment.

The material in this International Standard is technically oriented, and it is assumed that readers have at least a basic understanding of system and network security. The International Standard provides background information to help such readers understand the topics that are discussed. Readers are encouraged to take advantage of other resources (including those listed in this International Standard) for more detailed information.

Information Technology— BIOS Protection Guidelines

1 Scope

This International Standard provides requirements and guidelines for preventing the unauthorized modification of *Basic Input/Output System (BIOS)* firmware on PC client systems. Unauthorized modification of BIOS firmware by malicious software constitutes a significant threat because of the BIOS's unique and privileged position within the PC architecture. A malicious BIOS modification could be part of a sophisticated, targeted attack on an organization —either a permanent denial of service (if the BIOS is corrupted) or a persistent malware presence (if the BIOS is implanted with malware).

As used in this publication, the term BIOS refers to conventional BIOS, *Extensible Firmware Interface (EFI)* BIOS, and *Unified Extensible Firmware Interface (UEFI)* BIOS. This International Standard applies to system BIOS firmware (e.g., conventional BIOS or UEFI BIOS) stored in the system flash memory of computer systems, including portions that may be formatted as Option ROMs. However, it does not apply to Option ROMs, UEFI drivers, and firmware stored elsewhere in a computer system.

Subclause 7.2 provides platform vendors with requirements for a secure BIOS update process. Additionally, subclause 7.3 provides guidelines for managing the BIOS in an operational environment.

While this International Standard focuses on current and future x86 and x64 client platforms, the controls and procedures are independent of any particular system design.

2 Conformance

The following terms are used in this standard to indicate mandatory requirements, recommended options, or permissible actions.

- The terms "shall" and "shall not" indicate requirements to be followed strictly in order to conform to this standard and from which no deviation is permitted.
- The terms "should" and "should not" indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited.
- The terms "may" and "need not" indicate a course of action permissible within the limits of this standard.

An implementation is conformant to this standard if it implements the requirements specified in subclause 7.2.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

FIPS 186-4, *Digital Signature Standard.* July 2013.

NIST SP 800-89, *Recommendation for Obtaining Assurances for Digital Signature Applications.* November 2006.

NIST SP 800-131A, *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*. January 2011.

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1
Basic Input/Output System (BIOS)
boot firmware, such as those based on the conventional BIOS, Extensible Firmware Interface (EFI), and the Unified Extensible Firmware Interface (UEFI)

4.2
conventional BIOS
legacy boot firmware used in many x86-compatible computer systems (also known as the legacy BIOS)

4.3
Core Root of Trust for Measurement (CRTM)
the first piece of BIOS code that executes on the main processor during the boot process. On a system with a Trusted Platform Module the CRTM is implicitly trusted to bootstrap the process of building a measurement chain for subsequent attestation of other firmware and software that is executed on the computer system.

4.4
Extensible Firmware Interface (EFI)
a specification for the interface between the operating system and the platform firmware. Version 1.10 of the EFI specifications was the final version of the EFI specifications, and subsequent revisions made by the Unified EFI Forum are part of the UEFI specifications

4.5
firmware
software that is included in read-only memory (ROM)

4.6
option ROM
firmware that is called by the system BIOS, such as BIOS firmware on add-on cards (e.g., video card, hard drive controller, network card) as well as modules which extend the capabilities of the system BIOS

4.7
Protected Mode
an operational mode found in x86-compatible processors with hardware support for memory protection, virtual memory, and multitasking

4.8
Real Mode
a legacy high-privilege operating mode in x86-compatible processors

4.9
System Management Mode (SMM)
a high-privilege operating mode found in x86-compatible processors used for low-level system management functions

4.10
system flash memory
the non-volatile storage location of system BIOS, typically in electronically erasable programmable read-only memory (EEPROM) flash memory on the motherboard. While system flash memory is a technology-specific term, requirements and guidelines in this document referring to the system flash memory are intended to apply to any non-volatile storage medium containing the system BIOS.

4.11
Trusted Platform Module (TPM)
a tamper-resistant integrated circuit built into some computer motherboards that can perform cryptographic operations (including key generation) and protect small amounts of sensitive information, such as passwords and cryptographic keys

4.12
Unified Extensible Firmware Interface (UEFI)
a specification for the interface between the operating system and the platform firmware developed by the UEFI Forum

5 Symbols (and abbreviated terms)

ACPI
Advanced Configuration and Power Interface

BDS
Boot Device Selection

BIOS
Basic Input/Output System

CPU
Central Processing Unit

CRTM
Core Root of Trust for Measurement

DXE
Driver Execution Environment

EEPROM
Electrically Erasable Programmable Read-Only Memory

EFI
Extensible Firmware Interface

FIPS
Federal Information Processing Standard

GPT
GUID Partition Table

GUID
Globally Unique Identifier

MBR
Master Boot Record

OEM
Original Equipment Manufacturer

OS
Operating System

PEI
Pre-EFI Initialization

POST
Power-on self-test

PXE
Preboot Execution Environment

ROM
Read-only Memory

RT
Runtime

RTU
Root of Trust for Update

SMI
System Management Interrupt

SMM
System Management Mode

TPM
Trusted Platform Module

UEFI
Unified Extensible Firmware Interface

6 Background

6.1 System BIOS

The system BIOS is the first piece of software executed on the main central processing unit (CPU) when a computer is powered on. While the system BIOS was originally responsible for providing operating systems access to hardware, its primary role on modern machines is to initialize and test hardware components and load the operating system. In addition, the BIOS loads and initializes important system management functions, such as power and thermal management. The system BIOS may also load CPU microcode patches during the boot process.

There are several different types of BIOS firmware. Some computers use a 16-bit conventional BIOS, while many newer systems use boot firmware based on the UEFI specifications [23]. In this International Standard we refer to all types of boot firmware as BIOS firmware, the system BIOS, or simply BIOS. When necessary, we differentiate conventional BIOS firmware from UEFI firmware by calling them the conventional BIOS and UEFI BIOS, respectively.

System BIOS is typically developed by both original equipment manufacturers (OEMs) and independent BIOS vendors, and is distributed to end users with computer hardware. Manufacturers frequently update

system firmware to fix bugs, patch vulnerabilities, and support new hardware. The system BIOS is typically stored on electrically erasable programmable read-only memory (EEPROM) or other forms of flash memory, and is modifiable by end users. Typically, system BIOS firmware is updated using a utility or tool that has special knowledge of the non-volatile storage components in which the BIOS is stored.

A given computer system can have BIOS in several different locations. In addition to the motherboard, BIOS can be found on hard drive controllers, video cards, network cards and other add-in cards. This additional firmware generally takes the form of *Option ROMs* (containing conventional BIOS and/or UEFI drivers). These are loaded and executed by the system firmware during the boot process. Other system devices, such as hard drives and optical drives, may have their own microcontrollers and other types of firmware.

As noted in clause 1, the requirements and guidelines in this International Standard apply to BIOS firmware stored in the system flash. This includes Option ROMs and UEFI drivers that are stored with the system BIOS firmware and are updated by the same mechanism. It does not apply to Option ROMs, UEFI drivers, and firmware stored elsewhere in a computer system.

6.2 Role of system BIOS in the boot process

The primary function of the system BIOS is to initialize important hardware components and to load the operating system. This process is known as *booting*. The boot process of the system BIOS typically executes in the following stages:

1. Execute Core Root of Trust: The system BIOS may include a small core block of firmware that executes first and is capable of verifying the integrity of other firmware components. This has traditionally been called the *BIOS Boot Block*. For trusted computing applications, it may also contain the Core Root of Trust for Measurement (CRTM).

2. Initialize and Test Low-Level Hardware: Very early in the boot process the system BIOS initializes and tests key pieces of hardware on the computer system, including the motherboard, chipset, memory and CPU.

3. Load and Execute Additional Firmware Modules: The system BIOS executes additional pieces of firmware that either extend the capabilities of the system BIOS or initialize other hardware components necessary for booting the system. These additional modules may be stored within the same flash memory as the system BIOS or they may be stored in the hardware devices they initialize (e.g., video card, local area network card).

4. Select Boot Device: After system hardware has been configured, the system BIOS searches for a boot device (e.g., hard drive, optical drive, USB drive) and executes the boot loader stored on that device.

5. Load Operating System: While the system BIOS is still in control of the computer, the boot loader begins to load and initialize the operating system kernel. Once the kernel is functional, primary control of the computer system transfers from the system BIOS to the operating system.

In addition, the system BIOS loads system management interrupt (SMI) handlers (also known as System Management Mode (SMM) code) and initializes Advanced Configuration and Power Interface (ACPI) tables and code. These provide important system management functions for the running computer system, such as power and thermal management.

This clause describes the boot process in conventional BIOS-based systems and the boot process in UEFI-based systems. While conventional BIOS is used in many desktop and laptop computers deployed today, the industry has begun transitioning to UEFI BIOS.

6.2.1 Conventional BIOS boot process

Figure 1 shows a typical boot process for x86-compatible systems running a conventional BIOS. The conventional BIOS often executes in 16-bit real mode, although some more recent implementations execute

in protected mode. Some conventional BIOS-based firmware has a small block of BIOS firmware— known as the BIOS boot block— that is logically separate from the rest of the BIOS. On these computer systems, the boot block is the first firmware executed during the boot process. The boot block is responsible for checking the integrity of the remaining BIOS code, and may provide mechanisms for recovery if the main system BIOS firmware is corrupted. On most trusted computing architectures, the BIOS boot block serves as the computer system's CRTM because this firmware is implicitly trusted to bootstrap the process of building a measurement chain for subsequent attestation of other firmware and software that is executed on the machine [20].

The boot block executes the part of the conventional BIOS that initializes most hardware components—the *Power-on-Self-Test* (POST) code. During POST, key low-level hardware on the computer system is initialized, including the chipset, CPU, and memory. The system BIOS initializes the video card, which may load and execute its own BIOS to initialize graphics processors and memory.

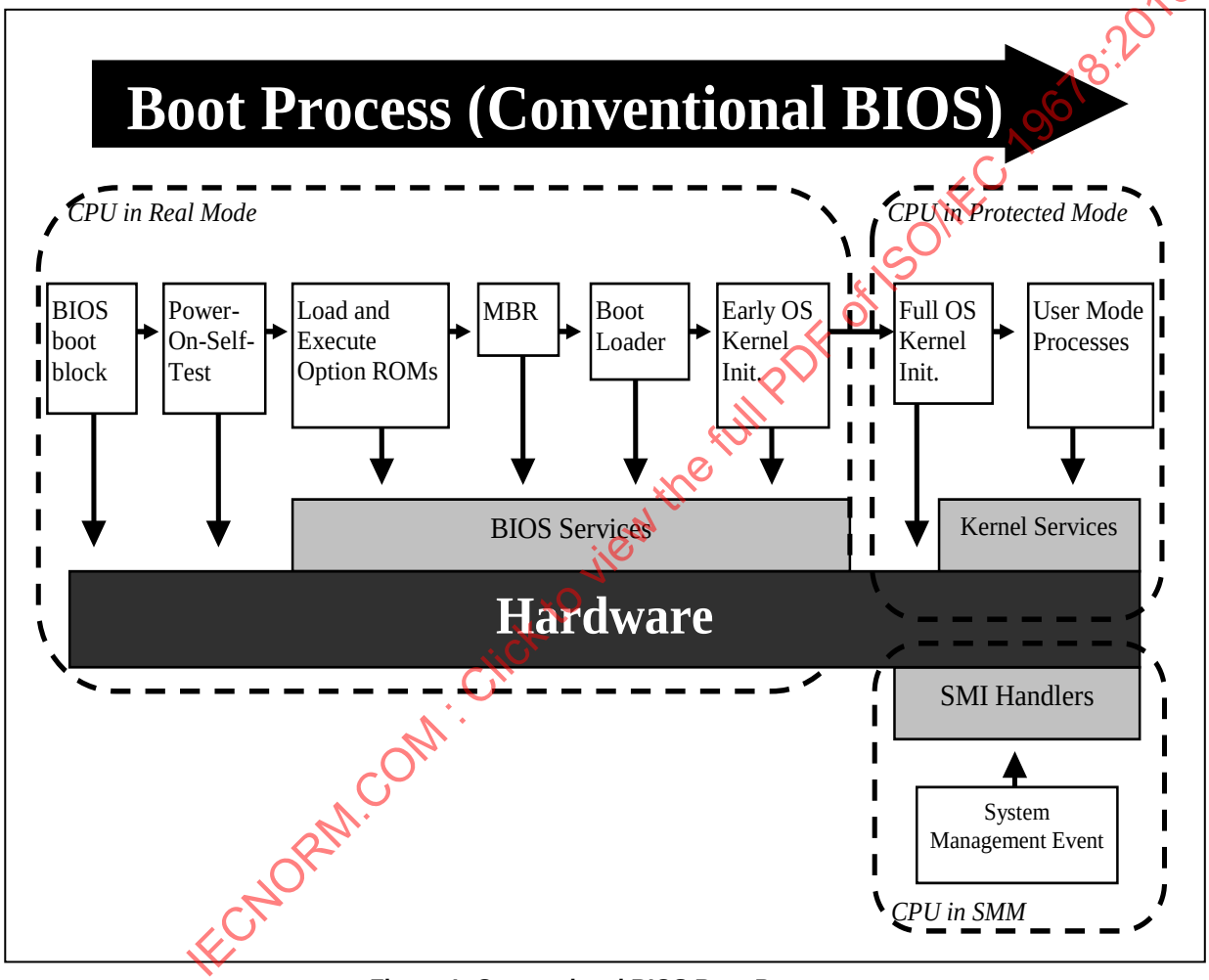

Figure 1: Conventional BIOS Boot Process

Next, the system BIOS searches for other peripherals and microcontrollers, and executes any Option ROMs on these components necessary to initialize them. Option ROMs execute very early in the boot process and can add a variety of features to the boot process. For example, the Option ROM on a network adapter could load the Preboot Execution Environment (PXE), which allows a computer to boot over the network.

Next, the system BIOS scans the computer system for storage devices that have been identified as boot devices. In a typical case, the BIOS attempts to boot from the first boot device it finds that has a valid master boot record (MBR). The MBR points to a boot loader stored on the hard drive, which in turn starts the process of loading the operating system.

During the boot process the system BIOS loads SMI handlers and initializes ACPI tables and code. SMI handlers run in a special high-privilege mode on the CPU known as System Management Mode, a 32-bit mode that is capable of bypassing many of the hardware security mechanisms of protected mode, such as memory segmentation and page protections.

6.2.2 UEFI boot process

At a high level, the UEFI boot process, shown in Figure 2, follows a similar flow to the conventional BIOS boot process. One difference is that UEFI code runs in 32- or 64-bit protected mode on the CPU, not in 16-bit real mode as is often the case with conventional BIOS. Most UEFI-based platforms start with a small core block of code that has the primary responsibility of authenticating subsequent code executed on the computer system. This is very similar to the role of the boot block in conventional BIOS. This part of the boot process is known as the Security (SEC) phase, and it serves as the core root of trust in the computer system.

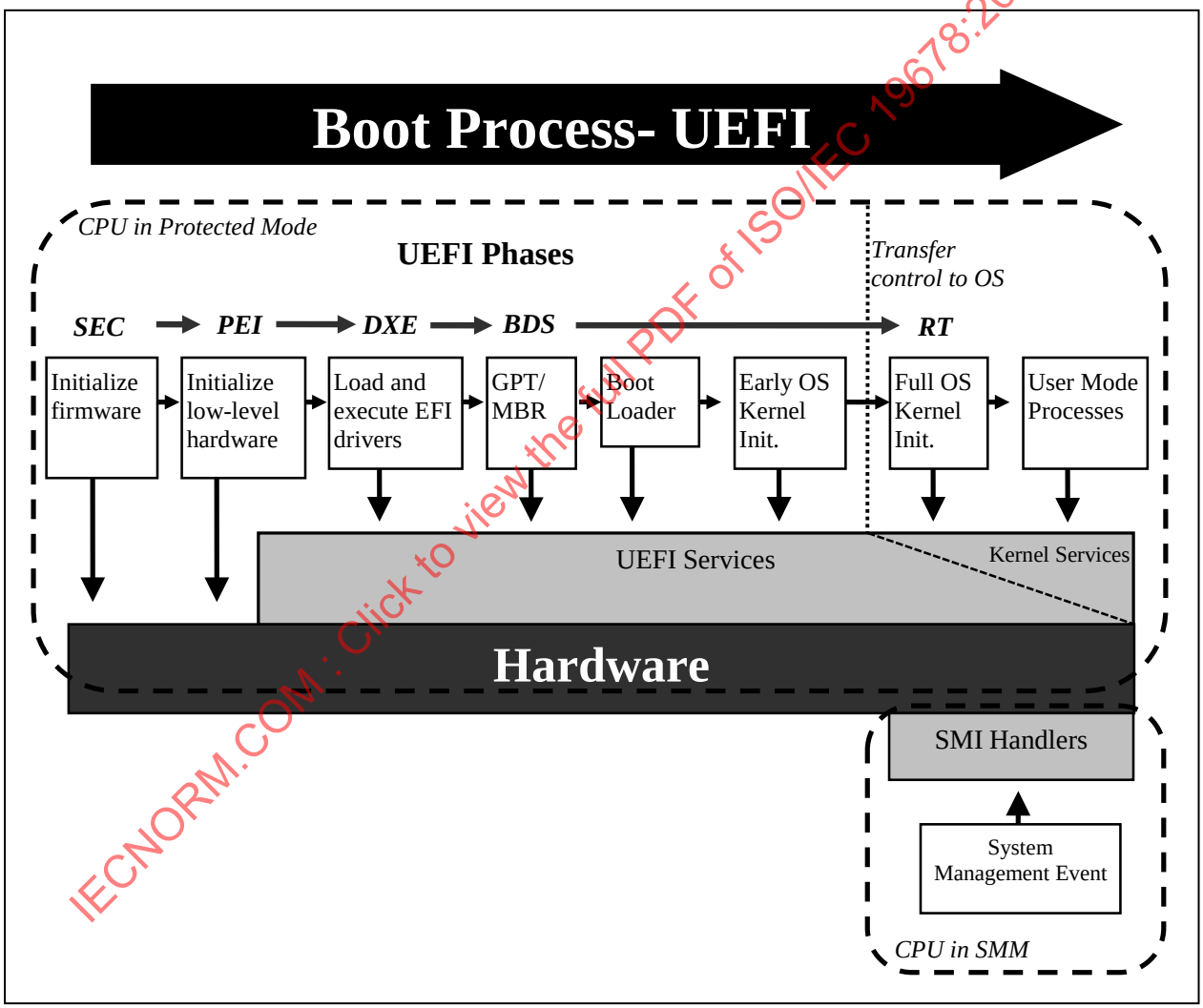

Figure 2: UEFI Boot Process

The next phase of the UEFI boot process is the Pre-EFI Initialization (PEI) Phase. The PEI phase is intended to initialize key system components, such as the processor, chipset and motherboard. In some cases, the code in the Security Phase and the PEI Phase comprise the core root of trust in a UEFI system.

The purpose of the PEI Phase is to prepare the system for the Driver Execution Environment (DXE) phase. The DXE phase is where most system initialization is performed. The firmware executed in this phase is responsible for searching for and executing drivers that provide device support during the boot process, or

provide additional features. During this phase the UEFI BIOS may execute conventional option ROMs, which have a similar purpose.

The PEI and DXE phases of the UEFI boot process lay the foundation to load an operating system. The final tasks necessary to load an operating system are performed in the Boot Device Selection (BDS) phase. This phase initializes console devices for simple input/output operations on the system. These console devices include local text or graphical interfaces, as well as remote interfaces, such as Telnet or remote displays over HTTP. The BDS phase also loads any additional drivers necessary to manage console or boot devices. Finally, the firmware loads the boot loader from the first MBR or GUID Partition Table (GPT) formatted boot device, and loads the operating system.

During the boot process the UEFI BIOS loads SMI handlers and initializes ACPI tables and code.

The Run Time phase of the UEFI boot process begins when the operating system is ready to take control from the UEFI BIOS. UEFI runtime services are available to the operating system during this phase.

6.3 Updating the system BIOS

A system and its supporting management software and firmware may provide several authorized mechanisms for legitimately updating the system BIOS. These include:

1. User-Initiated Updates**:** System and motherboard manufacturers typically supply end users with utilities capable of updating the system BIOS. Historically, end users booted from external media to perform these updates, but today most manufacturers provide utilities that can update the system BIOS from the user's normal operating system. Depending on the security mechanisms implemented on the system, these utilities might directly update the system BIOS or they may schedule an update for the next system reboot.
2. Managed Updates**:** A given computer system may have hardware and software-based agents that allow a system administrator to remotely update the system BIOS without direct involvement from the user.
3. Rollback**:** System BIOS implementations that authenticate updates before applying them may also check version numbers during the update process. In these cases, the system BIOS may have a special update process for rolling back the installed firmware to an earlier version. For instance, the rollback process might require the physical presence of the user. This mechanism guards against attackers flashing old firmware with known vulnerabilities.
4. Manual Recovery**:** To recover from a corrupt or malfunctioning system BIOS, many computer systems provide mechanisms to allow a user with physical presence during the boot process to replace the current system BIOS with a known good version and configuration.
5. Automatic Recovery**:** Some computer systems are able to detect when the system BIOS has been corrupted and recover from a backup firmware image stored in a separate storage location from the primary system BIOS (e.g., a second flash memory chip, a hidden partition on a hard drive).

6.4 Importance of BIOS integrity

As the first code that is executed by the main CPU, the system BIOS is a critical security component of a computer system. While the system BIOS, possibly with the use of a Trusted Platform Module (TPM), can verify the integrity of firmware and software executed later in the boot process, typically all or part of the system BIOS is implicitly trusted.

The system BIOS is a potentially attractive target for attack. Malicious code running at the BIOS level could have a great deal of control over a computer system. It could be used to compromise any components that are loaded later in the boot process, including the SMM code, boot loader, hypervisor, and operating system. The BIOS is stored on non-volatile memory that persists between power cycles. Malware written into a BIOS could be used to re-infect machines even after new operating systems have been installed or hard drives replaced. Because the system BIOS runs early in the boot process with very high privileges on the machine, malware running at the BIOS level may be very difficult to detect. Because the BIOS loads first, there is no opportunity for anti-malware products to authoritatively scan the BIOS.

BIOS exploits would likely be highly system-specific—directed at a specific version of a system BIOS or certain hardware components (e.g., a particular motherboard chipset). In contrast, most malware targets software executing at or above the operating system kernel, where it is easier to develop and can attack larger classes of machines. BIOS-level malware may be more likely employed in targeted attacks on high-value computer systems. The move to UEFI-based BIOS may make it easier for malware to target the BIOS in a widespread fashion, as these BIOS implementations are based on a common specification.

For the reasons outlined above, there are few known instances of BIOS-level malware. At this time, the only publicly known malware targeting the system BIOS that has infected a significant number of computers is the CIH virus, also known as the Chernobyl virus [19], first discovered in 1998. One element of the payload of this virus attempted to overwrite the BIOS on systems using a specific chipset that was widely deployed at the time. This malware relied on several vulnerabilities that are not present in modern machines.

Security researchers have demonstrated other potential attacks on conventional BIOS and EFI/UEFI firmware. Proof-of-concept attacks have been demonstrated that allow for the insertion of malicious code into conventional BIOS implementations that permit unsigned updates [13]. Other researchers have discovered a buffer-overflow vulnerability in the EFI BIOS on a modern platform. Although this EFI BIOS write-protects firmware early in the boot process and only flashes signed updates to firmware, the buffer overflow allowed the researchers to bypass the secure update process by executing an unsigned portion of the firmware update package before write protections were applied [23].

Vulnerabilities such as these could allow attackers to create stealthy malware that operate with very high privileges on a system. The system BIOS loads SMI handlers before passing control of the computer to the operating system. Malicious code written into a BIOS could modify the SMI handlers to create malware that would run in SMM [3]. This would give the malware unrestricted access to physical memory and peripherals connected to the host machine, and it would be very difficult for software running on the operating system to detect.

6.5 Threats to the system BIOS

The preceding clause established the importance of maintaining the integrity of the system BIOS. This clause describes some of the various ways that the integrity of the system BIOS can be attacked, and identifies the attacks considered within scope for the security controls and processes specified in clause 7.

The first threat to the integrity of the system BIOS comes while the system moves through the supply chain. Supply chain security techniques are out of scope for the security controls specified in this International Standard. Some of the procedures specified in subclause 7.3 can, however, be used to identify and remedy systems that have an unapproved system BIOS.

Assuming that the system arrives with the manufacturer's intended system BIOS installed, there are a number of threats to the integrity of the system BIOS during the system's lifetime:

- One of the most difficult threats to prevent is a user-initiated installation of a malicious system BIOS. User-initiated BIOS update utilities are often the primary method for updating the system BIOS. The requirements and guidelines included in this International Standard will not prevent users from installing unapproved BIOS images if they have physical access to the computer system. As with supply chain threats, security processes may be able to detect and remediate the unapproved system BIOS, such as initiating a recovery process to restore to an approved BIOS.
- Malware could leverage weak BIOS security controls or exploit vulnerabilities in the system BIOS itself to reflash or modify the system BIOS. General-purpose malicious software is unlikely to include this functionality, but a targeted attack on an organization could be directed towards an organization's standard system BIOS. The malicious BIOS can be delivered to the system either over a network, or using media. The requirements and guidelines presented in this International Standard are designed to prevent these kinds of attack.
- Network-based system management tools could also be used to launch an organization-wide attack on system BIOSs. For example, consider an organization-maintained update server for the organization's deployed system BIOS; a compromised server could push a malicious system BIOS to computer systems across the organization. This is a high-impact attack, but requires either an

insider or compromise of an organization's update process. The requirements and guidelines presented in this International Standard are designed to prevent this kind of attack.

- Any of the preceding mechanisms could be used to rollback to an authentic but vulnerable system BIOS. This is a particularly insidious attack, since the "bad" BIOS is authentic (i.e., shipped by the manufacturer). The security controls specified in the following clause are primarily focused on verifying the source and integrity of the system BIOS. This International Standard includes a recommendation for rollback protection.

The controls described in the following clause are primarily focused on preventing unauthorized modification of the system BIOS by potentially malicious software running on computer systems. Installation of an unapproved system BIOS in the supply chain, by individuals with physical access, or through rollback to an authenticated but vulnerable system BIOS, are not addressed by the requirements in subclause 7.2, but can be addressed using processes specified in subclause 7.3.

7 Threat mitigation

7.1 Overview

BIOS is a critical component of a secure system. As the first code executed during the boot process, the system BIOS is implicitly trusted by hardware and software components in a system. The previous clause described the system BIOS's role in the boot process, the system BIOS's appeal to attackers, and the potential threats resulting in the unauthorized modification of the BIOS. This clause presents security requirements for BIOS implementations and recommended practices for managing BIOSs in an enterprise environment. Subclause 7.2 provides requirements for a secure BIOS update process. It is intended for platform vendors designing, implementing, or selecting a system BIOS implementation. While products may not be immediately available, organizations can use these requirements at input to their procurement processes and begin developing plans to make use of these security features when they are available. Organizations can use the recommended BIOS management practices in subclause 7.3 when developing these plans. The recommendations are intended to prevent unauthorized modification of the BIOS.

7.2 Security requirements for system BIOS implementations

This subclause provides requirements intended to maintain the integrity of the BIOS after it has been provisioned by securing the mechanisms used for updating the BIOS. In particular, this subclause defines requirements for system BIOS implementations for a secure BIOS update mechanism. A secure BIOS update mechanism includes:

1. a process for verifying the authenticity and integrity of BIOS updates; and
2. a mechanism for ensuring that the BIOS is protected from modification outside of the secure update process.

Authentication verifies that a BIOS update image was generated by an authentic source and is unaltered. All updates to the system BIOS shall either go through an authenticated BIOS update mechanism as described in subclause 7.2.1 or use an optional secure local update mechanism compliant with the requirements in subclause 7.2.2.

These requirements for a secure BIOS update mechanism do not mitigate all risks associated with the system BIOS. Some threats to unauthorized modification of the system BIOS remain. For example, these requirements do not prevent individuals with physical access to systems from modifying the system BIOS. Nor do they guarantee the absence of vulnerabilities in the system BIOS implementations. The requirements on the system BIOS should be used in conjunction with organizations' existing security policies and procedures.

7.2.1 BIOS update authentication

The authenticated BIOS update mechanism employs digital signatures to ensure the authenticity of the BIOS update image. To update the BIOS using the authenticated BIOS update mechanism, there shall be a Root of Trust for Update (RTU) that contains a signature verification algorithm and a key store that includes the public key needed to verify the signature on the BIOS update image. The key store and the signature verification algorithm shall be stored in a protected fashion on the computer system and shall be

modifiable only using an authenticated update mechanism or a secure local update mechanism as outlined in subclause 7.2.2.

The key store in the RTU shall include a public key used to verify the signature on a BIOS update image or include a cryptographic hash of the public key if a copy of the public key is provided with the BIOS update image. In the latter case, the update mechanism shall hash the public key provided with the BIOS update image and ensure that it matches a hash which appears in the key store before using the provided public key to verify the signature on the BIOS update image.

BIOS images shall be signed in conformance with NIST SP 800-89, *Recommendation for Obtaining Assurances for Digital Signature Applications*, using an approved digital signature algorithm as specified in NIST FIPS 186-3, *Digital Signature Standard*, that provides at least 112 bits of security strength, in accordance with NIST SP 800-131A, *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*.

The update mechanism shall ensure that the BIOS update image has been digitally signed and that the digital signature can be verified using a key in the RTU before updating the BIOS. Recovery mechanisms shall also use the authenticated update mechanism unless the recovery process meets the requirements for a secure local update. The authenticated update mechanism should prevent the unauthorized rollback of the BIOS to an earlier authentic version that has a known security weakness. This limitation of the rollback mechanism may be accomplished, for example, by verifying that the version number of the BIOS image is larger than the currently installed BIOS image's version number.

Some organizations may wish to assert greater control over BIOS updates in high-security environments. The authenticated update mechanism may be designed to permit organizational control over the update process, where updates to the BIOS or rollbacks of the BIOS to an earlier version are permitted only if the update or rollback has been authorized by the organization. For example, specific BIOS images could be authorized by an organization by countersigning them with an organization-controlled key, which would be verified during the update process.

7.2.2 Secure local update

BIOS implementations may optionally include a secure local update mechanism that updates the system BIOS without using the authenticated update mechanism. The secure local update mechanism, if it is implemented, should be used only to load the first BIOS image or to recover from a corruption of a system BIOS that cannot be fixed using the authenticated update mechanism described in subclause 7.2.1. A secure local update mechanism shall ensure the authenticity and integrity of the BIOS update image by requiring physical presence. Further protections may be implemented in the secure local update mechanism by requiring the entry of an administrator password or the unlocking of a physical lock (e.g., a motherboard jumper) before permitting the system BIOS to be updated.

7.2.3 Integrity protection

To prevent unintended or malicious modification of the system BIOS outside the authenticated BIOS update process, the RTU and the system BIOS (excluding configuration data used by the system BIOS that is stored in non-volatile memory) shall be protected from unintended or malicious modification with a mechanism that cannot be overridden outside of an authenticated BIOS update. The protection mechanism shall itself be protected from unauthorized modification.

The authenticated BIOS update mechanism shall be protected from unintended or malicious modification by a mechanism that is at least as strong as that protecting the RTU and the system BIOS.

The protection mechanism shall protect relevant regions of the system flash memory containing the system BIOS prior to executing firmware or software that can be modified without using an authenticated update mechanism or a secure local update mechanism. Protections should be enforced by hardware mechanisms that are not alterable except by an authorized mechanism.

7.2.4 Non-bypassability