BASIC INPUT/OUTPUT SYSTEMS

Aparajita, Diksha
I.T.-1, Dronacharya College Of Engg.

***Abstract*- Our research paper is on the topic " Basic Input/output Systems". Our topic covers the details on Input/Output hardware, application input/output interface, kernel, transforming input/output requests, performance issues and threads. Input/output devices are used by humans to communicate with the system and give instructions to it. In this paper, we will provide you with the details of a lot of architecture involved in this inter-connection. Whenever a processor does its work, an input/output interface is a must. There are certain logics which are pre-requisitely required in this. Also, a great deal of programming and de-bugging is required. Our research covers all these topics verily certain issues and complications arise in any system. The certain types of input/output systems, their future prospects involved and the solutions present every-time.**

I. 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 document provides security guidelines 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 document 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 guidelines 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.

II. HISTORY

The term BIOS (Basic Input/output System) was invented by Gary Kendal and first appeared in the CP/M operating system in 1975, describing the machine-specific part of CP/M loaded during boot time that interfaces directly with the hardware.[3] (A CP/M machine usually has only a simple boot loader in its ROM.)

Versions of MS-DOS, PC DOS or DR-DOS contain a file called variously "IO.SYS", "IBMBIO.COM", "IBMBIO.SYS", or "DRBIOS.SYS"; this file is known as the "DOS BIOS" and contains the lower-level hardware-specific part of the operating system. Together with the underlying hardware-specific, but operating system-independent "System BIOS", which resides in ROM, it represents the analogous to the "CP/M BIOS".

With the introduction of PS/2 machines, IBM divided the System BIOS into real-mode and protected mode portions. The real-mode portion was meant to provide backward-compatibility with existing operating systems such as DOS, and therefore was named "CBIOS" (for Compatibility BIOS), whereas the "ABIOS" (for Advanced BIOS) provided new interfaces specifically suited for multitasking operating systems such as OS/2.

BIOS user interface

The BIOS of the original IBM PC XT had no interactive user interface. Error codes or messages were displayed on the screen, or coded series of

sounds were generated to signal errors (when the POST had not proceeded to the point of successfully initializing a video display adapter). Options on the PC and XT were set by switches and jumpers on the main board and on peripheral cards. Starting around the mid-1990s, it became typical for the BIOS ROM to include a "BIOS configuration utility" or "BIOS setup utility", accessed at system power-up by a particular key sequence. This program allowed the user to set system configuration options, of the type formerly set using DIP switches, through an interactive menu system controlled through the keyboard. In the interim period, IBM-compatible PCs—including the IBM AT—held configuration settings in battery-backed RAM and used a bootable configuration program on disk, not in the ROM, to set the configuration options contained in this memory. The disk was supplied with the computer, and if it was lost the system settings could not be changed.

A modern Wintel-compatible computer provides a setup routine essentially unchanged in nature from the ROM-resident BIOS setup utilities of the late 1990s; the user can configure hardware options using the keyboard and video display. Also, when errors occur at boot time, a modern BIOS usually displays user-friendly error messages, often presented as pop-up boxes in a TUI style, and offers to enter the BIOS setup utility or to ignore the error and proceed if possible. Instead of battery-backed RAM, the modern Wintel machine may store the BIOS configuration settings in flash ROM, perhaps the same flash ROM that holds the BIOS itself.

Because it is the only visible feature of the BIOS to the average user, who is not familiar with hardware programming techniques and device abstraction layers, he often misidentifies the BIOS configuration utility as the BIOS, as in typical statements such as, "To add a second internal hard disk, you have to go into your BIOS and enable it," or, "To change the boot password you have to use the BIOS." This misuse of terms has become so common that it may now be considered that "BIOS configuration utility" or "BIOS setup menu" is a new second definition of the word "BIOS".

EEPROM chips are advantageous because they can be easily updated by the user; hardware manufacturers frequently issue BIOS updates to upgrade their products, improve compatibility and remove bugs. However, this advantage had the risk that an improperly executed or aborted BIOS update could render the computer or device unusable. To avoid these situations, more recent BIOSs use a "boot block"; a portion of the BIOS which runs first and must be updated separately. This code verifies if the rest of the BIOS is intact before transferring control to it. If the boot block detects any corruption in the main BIOS, it will typically warn the user that a recovery process must be initiated by booting from removable media (floppy, CD or USB memory) so the user can try flashing the BIOS again. Some motherboards have a BIOS (sometimes referred to as Dual BIOS boards) to recover from BIOS corruptions.

There are at least four known BIOS attack viruses, two of which were for demonstration purposes. The first one found in the wild was Maroni, targeting Chinese users.

The first BIOS virus was CIH, whose name matches the initials of its creator, Chen Ing Hau. CIH was also called the "Chernobyl Virus", because its payload date was 1999-04-26, the 13th anniversary of the Chernobyl accident. CIH appeared in mid-1998 and became active in April 1999. It was able to erase flash ROM BIOS content. Often, infected computers could no longer boot, and people had to remove the flash ROM IC from the motherboard and reprogram it. CIH targeted the then-widespread Intel i430TX motherboard chipset and took advantage of the fact that the Windows 9x operating systems, also widespread at the time, allowed direct hardware access to all programs.

Modern systems are not vulnerable to CIH because of a variety of chipsets being used which are incompatible with the Intel i430TX chipset, and also other flash ROM IC types. There is also extra protection from accidental BIOS rewrites in the form of boot blocks which are protected from accidental overwrite or dual and quad BIOS equipped systems which may, in the event of a crash, use a backup BIOS. Also, all modern operating systems such as FreeBSD, Linux, OS X, Windows NT-based Windows OS like Windows 2000, Windows XP and newer, do not allow user-mode programs to have direct hardware access.

As a result, as of 2008, CIH has become essentially harmless, at worst causing annoyance by infecting executable files and triggering antivirus software. Other BIOS viruses remain possible,

however; since most Windows home users without Windows Vista/7's UAC run all applications with administrative privileges, a modern CIH-like virus could in principle still gain access to hardware without first using an exploit. The operating system Open BSD prevents all users from having this access and the grsecurity patch for the linux kernel also prevents this direct hardware access by default, the difference being an attacker requiring a much more difficult kernel level exploit or reboot of the machine.

The second BIOS virus was a technique presented by John Heisman, principal security consultant for UK-based Next-Generation Security Software. In 2006, at the Black Hat Security Conference, he showed how to elevate privileges and read physical memory, using malicious procedures that replaced normal ACPI functions stored in flash memory.

The third BIOS virus was a technique called "Persistent BIOS infection." It appeared in 2009 at the CanSecWest Security Conference in Vancouver, and at the Sysco Security Conference in Singapore. Researchers Anibal Sacco and Alfredo Ortega, from Core Security Technologies, demonstrated how to insert malicious code into the decompression routines in the BIOS, allowing for nearly full control of the PC at start-up, even before the operating system is booted. The proof-of-concept does not exploit a flaw in the BIOS implementation, but only involves the normal BIOS flashing procedures. Thus, it requires physical access to the machine, or for the user to be root. Despite these requirements, Ortega underlined the profound implications of his and Sacco's discovery: "We can patch a driver to drop a fully working root. We even have a little code that can remove or disable antivirus.

Mebromi is a Trojan which targets computers with Award BIOS, Microsoft Windows, and antivirus software from two Chinese companies: Rising Antivirus and Jiangxi KV Antivirus. Mebromi installs a root kit which infects the master boot record.

In a December 2013 interview with CBS 60 Minutes, Deborah Plunkett, Information Assurance Director for the US National Security Agency claimed that NSA analysts had uncovered and thwarted a possible BIOS attack by a foreign nation state. The attack on the world's computers could have allegedly "literally taken down the US economy." The segment further cites anonymous cyber security experts briefed on the operation as alleging the plot was conceived in China . A later article in The Guardian cast doubt on the likelihood of such a threat, quoting Berkeley computer-science researcher Nicholas Weaver, Matt Blaze, a computer and information sciences professor at the University of Pennsylvania, and cyber security expert Robert David Graham in an analysis of the NSA's claims.

III. ALTERNATIVES

New standards grafted onto the BIOS are usually without complete public documentation or any BIOS listings. As a result, it is not as easy to learn the intimate details about the many non-IBM additions to BIOS as about the core BIOS services. If the boot block detects any corruption in the main BIOS, it will typically warn the user that a recovery process must be initiated by booting from removable media (floppy, CD or USB memory) so the user can try flashing the BIOS again. Some motherboards have a *backup* BIOS (sometimes referred to as Dual BIOS boards) to recover from BIOS corruption. In other types of computers, the terms *boot monitor*, *boot loader*, and *boot ROM* may be used instead.

As of 2011, the BIOS is being replaced by the more complex Extensible Firmware Interface (EFI) in many new machines. EFI is a specification which replaces the runtime interface of the legacy BIOS. Initially written for the Itanium architecture, EFI is now available for x86 and x86-64 platforms; the specification development is driven by The Unified EFI Forum, an industry Special Interest Group. EFI booting has been supported in only Microsoft Windows versions supporting GPT, the Linux kernel 2.6.1 and later, and Mac OS X on Intel-based Macs.

Other alternatives to the functionality of the "Legacy BIOS" in the x86 world include core boot.

A number of larger, more powerful servers and workstations use a platform-independent Open Firmware (IEEE-1275) based on the programming language; it is included with Sun's SPARC computers, IBM's RS/6000 line, and other PowerPC systems such as the CHRP motherboards, along with the x86-based OLPC XO-1. Later x86-based personal computer operating systems, like Windows NT, use their own, native drivers; this makes it much easier to extend support to new hardware.

IV. SUMMARY

With BIOS, your operating system and its applications are freed from having to understand exact details (such as hardware addresses) about the attached input/output devices. When device details change, only the BIOS program needs to be changed. Sometimes this change can be made during your system setup. In any case, neither your operating system or any applications you use need to be changed.

Although BIOS is theoretically always the intermediary between the microprocessor and I/O device control information and data flow, in some cases, BIOS can arrange for data to flow directly to memory from devices (such as video cards) that require faster data flow to be effective.

BIOS is an integral part of your computer and comes with it when you bring it home.