Unified Extensible Firmware Interface

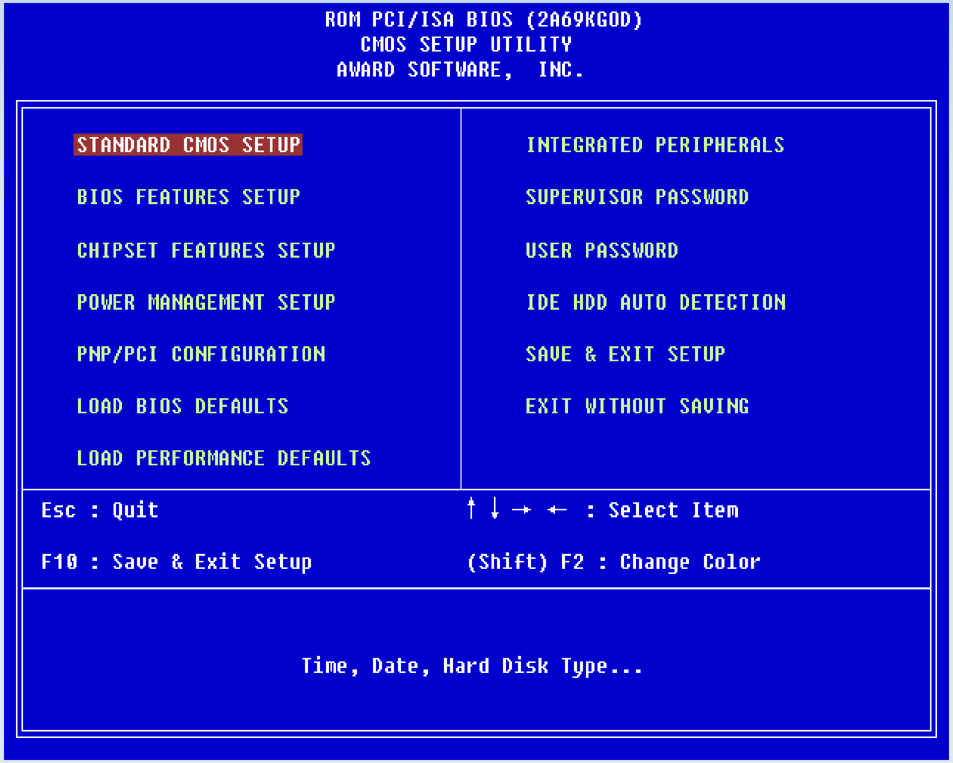
Presentation to Linux Users of Victoria on LUV's Policy on UEFI

December 6, 2011

http://levlafayette.com

Unified Extensible Firmware Interface

UEFI is a specification for a software interface between operating systems and firmware. It is designed to an alternative and replacement to BIOS (Basic Input/Output System), the first code run by a system, to identify system devices (e.g., video display, keyboard/mouse, HDD, optical drives etc), then it locates a select boot device and executes ("bootstraps") the operating system. BIOS software is stored on a non-volatile ROM chip on the motherboard.



```
                 ROM PCI/ISA BIOS (2A69KG0D)
                     CMOS SETUP UTILITY
                    AWARD SOFTWARE,  INC.

   STANDARD CMOS SETUP              INTEGRATED PERIPHERALS

   BIOS FEATURES SETUP              SUPERVISOR PASSWORD

   CHIPSET FEATURES SETUP           USER PASSWORD

   POWER MANAGEMENT SETUP           IDE HDD AUTO DETECTION

   PNP/PCI CONFIGURATION            SAVE & EXIT SETUP

   LOAD BIOS DEFAULTS               EXIT WITHOUT SAVING

   LOAD PERFORMANCE DEFAULTS

 Esc : Quit                        ↑ ↓ → ←  : Select Item
 F10 : Save & Exit Setup           (Shift) F2 : Change Color

                   Time, Date, Hard Disk Type...
```

Unified Extensible Firmware Interface

In 1998 PC BIOS limitations (e.g., 16 bit processor, 1 MB addressable memory) initiated the Intel Boot Initiative, which was re-branded the EFI and, in 2005 to UEFI, with the establishment of specification version 1.10 and the UEFI Forum, a non-profit which includes representatives from AMD, American Megatrends, Apple, Dell, HP, IBM, Insyde Software, Intel, Lenovo, Microsoft, and Phoenix Technologies. Version 2.1 (January 2007) added cryptography and network authentication. The current UEFI specification, v2.3.1, was approved in April 2011.

http://computer.howstuffworks.com/bios1.htm

http://www.uefi.org/about

Unified Extensible Firmware Interface

BIOS activity has changed over time. Older OS used BIOS for most i/o tasks. This became increasingly inefficient and many roles were taken over by the OS which had their own native drives (faster and more flexible!) whereas BIOS started managing things like power and thermal management, hot swapping etc.

The absolutely biggest advantage of a BIOS is that it's _so_ inconvenient and obviously oldfashioned, that you have to be crazy to want to do anything serious in it. Real mode, 16-bit code is actually an _advantage_ in that sense. People know how to treat it, and don't get any ideas about it being some grandiose framework for anything else than "just load the OS and get the hell out of there". (Linus Tolvards, July 2006)

http://kerneltrap.org/node/6884

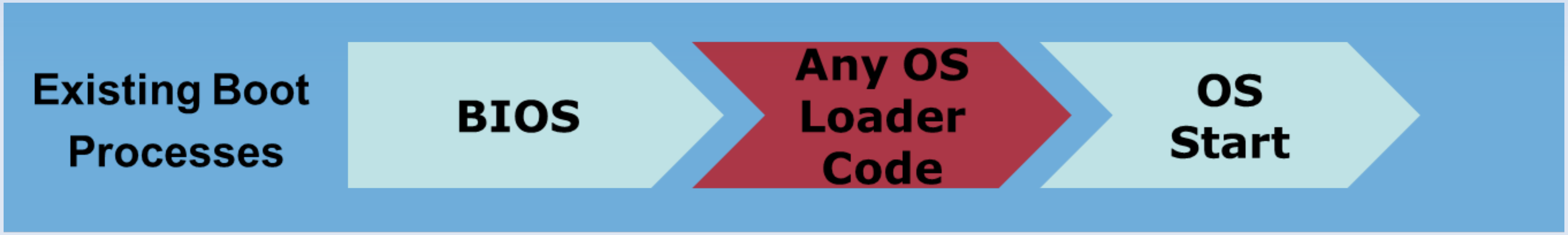
Unified Extensible Firmware Interface

BIOS is being replaced by UEFI; UEFI booting is supported by Microsoft OS products supporting GUID Partition Table (GPT), a replacement for the Master Boot Record partitioning scheme, for Mac in OS X for and for Linux using kernels 2.6.1 and greater via elilo and GRUB boot loaders. Also, the open source community is replacing for proprietary BIOSes through the coreboot and OpenBIOS/Open Firmware projects.

Apart from these technical limitations, BIOS-based systems are very suspectible to "bootkit" malware, typically where the legitimate boot loader is replaced, which can be used to attack full disk encryption systems, the malware persisting throgh the transition to protected mode when the kernel has loaded. One well known example is the TDL/Alureon rootkit (designed to search network traffic for usernames, passwords, credit card data etc), the second most active botnet in 2010, and described in The Registrar as "the world's most advanced rootkit", due to its ability to bypass the mandatory kernel-mode signing requirement in 64-bit Windows7.

Unified Extensible Firmware Interface

The latest version of UEFI includes a signed certificate module for hardware verification in the bootstrap process. Computers implementing UEFI secure boot will not be able to boot unauthorized operating systems, including initially authorised systems that have been modified (image from MS).



- The BIOS starts any OS loader, even malware

UEFI also will provide faster boot-time, ability to boot larger disks, CPU-independent architecture and drives and enhancements to existing PC BIOS features (e.g., Advanced Configuration and Power Interface (ACPI)). Collisions are avoided by operating system vendors registering a unique name, (e.g., a MS will not overwrite a Linux bootloader).

Unified Extensible Firmware Interface

What could possibly go wrong?!

Unified Extensible Firmware Interface

UEFI specifies a Platform Key (PK), which is supposed to be controlled by the Platform Owner (i.e., whoever owns the hardware) and a set of Key-Exchange Keys (KEKs), which are designed to be controlled by the OEM and OS vendors. These keys are public/private key pairs; whoever knows the private key is the key controller, but to install the key, you only need the public piece, which means KEKs can be installed by anybody without controlling them.

This it allows the hardware owner to decide which keys they trust without compromising the ability of the KEK controllers to assure themselves that the OS booted securely. Keys can also be added to a blacklist; binaries signed with a blacklisted key will not load.

Unified Extensible Firmware Interface

There is no centralised signing authority for these UEFI keys. If a vendor key is installed on a machine, the only way to get code signed with that key is to get the vendor to perform the signing. A machine may have several keys installed, but if you are unable to get any of them to sign your binary then it won't be installable.

Unified Extensible Firmware Interface

Microsoft has announced that if computer makers wish to distribute machines with the Windows 8 compatibility logo, they must have UEFI Secure Boot enabled. Certification does not require that the user be able to disable UEFI secure boot (which may remove the compatibility), and some hardware vendors will not have this option.

A system that ships with UEFI secure boot enabled and only includes Microsoft's signing keys will only securely boot Microsoft operating systems. Further, some Microsoft engineers have already suggested that end users may wish to give up control of their Platform Key (PK) to Microsoft and OEM suppliers, the holders of the Key-Exchange Keys (KEKs).

Steven Sinofsky "Protecting the pre-OS environment with UEFI"
http://blogs.msdn.com/b/b8/archive/2011/09/22/protecting-the-pre-os-environment-with-uefi.aspx

Unified Extensible Firmware Interface

"For Windows customers, Microsoft is using the Windows Certification program to ensure that systems shipping with Windows 8 have secure boot enabled by default, that firmware not allow programmatic control of secure boot (to prevent malware from disabling security policies in firmware), and that OEMs prevent unauthorized attempts at updating firmware that could compromise system integrity."

Re-engineering the Windows Boot Experience
http://blogs.msdn.com/b/b8/archive/2011/09/20/reengineering-the-windows-boot-experience.aspx

User control and ownership of hardware?
Coreboot and other minimal OS loaders?
Dual booting with Linux et al?

Unified Extensible Firmware Interface

A Brief and Very Incomplete Microsoft's history of anti-competitive behaviour

Consent decree for exclusionary licensing established in 1994, forcing PC manufacturers to pay for a copy even if the system didn't ship with MS-DOS. Attempted purchase of Intuit blocked on anti-competitive behaviour in 1994. Accused by Apple, Intel and the San Francisco Canyon Company of knowingly stealing several thousand lines of QuickTime source code in an effort to improve the performance of Video for Windows; settled in 1997 after threatening to withdraw support for Office for Mac. Finding of fact as an monopoly that establishes barrier to market entry by US Federal Court, 1999. Charged in 1999 by Caldera for deliberately modifying Windows 3.1 so that it would not run on DR DOS 6; settled out-of-court for an undisclosed sum. Ordered to pay Bristol Technologies in 2000 for $1 million after being accused of breaching an agreement after Bristol helped Microsoft develop server software. Found in 2004 to have violated non-disclosure agreements from Go Corporation several years prior. Fined €497 million for breach of EU competition laws, 2004. Settled a class-action suit for overcharging customers in 2004 for $1.1 billion, with an additional $258 million in legal fees. Fined for breaching competition law in South Korea, 2005. Fined for non-compliance with EU decision in 2008.

Unified Extensible Firmware Interface

The Linux Foundation, RedHat and Canonical and the Free Software Foundation have all released position and technical papers on these matters. The following represents a combination of the positions they have recommended:

* All OEMs allow Secure Boot to be easily disabled and enabled through a firmware configuration interface.
* All platforms that enable UEFI secure boot should ship in setup mode where the owner has control over which platform key (PK) is installed. It should also be possible for the owner to return a system to setup mode in the future if needed.
* The initial bootstrap of an operating system should detect a platform in the setup mode, install its own key-exchange key (KEK), and install a platform key to enable secure boot.
...

Unified Extensible Firmware Interface

...

* OEMs (with assistance from BIOS vendors) provide a standardised mechanism for configuring keys in system firmware
* A firmware-based mechanism should be established to allow a platform owner to add new key-exchange keys to a system running in secure mode so that dual-boot systems can be set up.
* A firmware-based mechanism for easily importing new keys from removable media.
* At some future time, an operating-system- and vendor-neutral certificate authority should be established to issue KEKs for third-party hardware and software vendors.

http://ozlabs.org/docs/uefi-secure-boot-impact-on-linux.pdf

http://www.linuxfoundation.org/publications/making-uefi-secure-boot-work-with-open-platforms

Unified Extensible Firmware Interface

Linux Users of Victoria encourage those who support and respect software freedom to add their name to the following statement from the Free Software Foundation:

We, the undersigned, urge all computer makers implementing UEFI's so-called "Secure Boot" to do it in a way that allows free software operating systems to be installed. To respect user freedom and truly protect user security, manufacturers must either allow computer owners to disable the boot restrictions, or provide a sure-fire way for them to install and run a free software operating system of their choice. We commit that we will neither purchase nor recommend computers that strip users of this critical freedom, and we will actively urge people in our communities to avoid such jailed systems.

http://www.fsf.org/campaigns/secure-boot-vs-restricted-boot/statement