Intel Management Engine (ME)

Introduced in June 2006 in Intel's 965 Express Chipset Family of (Graphics and) Memory Controller Hubs, or (G)MCHs, and the ICH8 I/O Controller Family, the Intel Management Engine (ME) is a separate computing environment physically located in the (G)MCH chip. In Q3 2009, the first generation of Intel Core i3/i5/i7 (Nehalem) CPUs and the 5 Series Chipset family of Platform Controller Hubs, or PCHs, brought a more tightly integrated ME (now at version 6.0) inside the PCH chip, which itself replaced the ICH. Thus, the ME is ***present on all Intel desktop, mobile (laptop), and server systems since mid 2006***.

The ME consists of an ARC processor core (replaced with other processor cores in later generations of the ME), code and data caches, a timer, and a secure internal bus to which additional devices are connected, including a cryptography engine, internal ROM and RAM, memory controllers, and a ***direct memory access (DMA) engine*** to access the host operating system's memory as well as to reserve a region of protected external memory to supplement the ME's limited internal RAM. The ME also has ***network access*** with its own MAC address through an Intel Gigabit Ethernet Controller. Its boot program, stored on the internal ROM, loads a firmware "manifest" from the PC's SPI flash chip. This manifest is ***signed with a strong cryptographic key***, which differs between versions of the ME firmware. If the manifest isn't signed by a specific Intel key, the boot ROM won't load and execute the firmware and the ME processor core will be halted.

The ME firmware is compressed and consists of modules that are listed in the manifest along with secure cryptographic hashes of their contents. One module is the operating system kernel, which is based on a ***proprietary real-time operating system (RTOS) kernel*** called "ThreadX". The developer, Express Logic, sells licenses and source code for ThreadX. Customers such as Intel are forbidden from disclosing or sublicensing the ThreadX source code. Another module is the Dynamic Application Loader (DAL), which consists of a ***Java virtual machine*** and set of preinstalled Java classes for cryptography, secure storage, etc. The DAL module can load and execute additional ME modules from the PC's HDD or SSD. The ME firmware also includes a number of native application modules within its flash memory space, including Intel Active Management Technology (AMT), an implementation of a Trusted Platform Module (TPM), Intel Boot Guard, and audio and video DRM systems.

The Active Management Technology (AMT) application, part of the Intel "vPro" brand, is a Web server and application code that enables remote users to power on, power off, view information about, and otherwise manage the PC. It can be ***used remotely even while the PC is powered off*** (via Wake-on-Lan). Traffic is encrypted using SSL/TLS libraries, but recall that all of the major SSL/TLS implementations have had highly publicized vulnerabilities. The AMT application itself has ***known vulnerabilities***, which have been exploited to develop rootkits and keyloggers and covertly gain encrypted access to the management features of a PC. Remember that the ME has full access to the PC's RAM. This means that an attacker exploiting any of these vulnerabilities may gain access to everything on the PC as it runs: all open files, all running applications, all keys pressed, and more.

Intel Boot Guard is an ME application introduced in Q2 2013 with ME firmware version 9.0 on 4th Generation Intel Core i3/i5/i7 (Haswell) CPUs. It allows a PC OEM to generate an asymmetric cryptographic keypair, install the public key in the CPU, and prevent the CPU from executing boot firmware that isn't signed with their private key. This means that ***coreboot and libreboot are impossible to port*** to such PCs, without the OEM's private signing key. Note that systems assembled from separately purchased mainboard and CPU parts are unaffected, since the vendor of the mainboard (on which the boot firmware is stored) can't possibly affect the public key stored on the CPU.

ME firmware versions 4.0 and later (Intel 4 Series and later chipsets) include an ME application for ***audio and video DRM*** called “Protected Audio Video Path” (PAVP). The ME receives from the host operating system an encrypted media stream and encrypted key, decrypts the key, and sends the encrypted media decrypted key to the GPU, which then decrypts the media. PAVP is also used by another ME application to draw an authentication PIN pad directly onto the screen. In this usage, the PAVP application directly controls the graphics that appear on the PC’s screen in a way that the host OS cannot detect. ME firmware version 7.0 on PCHs with 2nd Generation Intel Core i3/i5/i7 (Sandy Bridge) CPUs replaces PAVP with a similar DRM application called “Intel Insider”. Like the AMT application, these DRM applications, which in themselves are defective by design, demonstrate the omnipotent capabilities of the ME: this hardware and its proprietary firmware can access and control everything that is in RAM and even ***everything that is shown on the screen***.

The Intel Management Engine with its proprietary firmware has complete access to and control over the PC: it can power on or shut down the PC, read all open files, examine all running applications, track all keys pressed and mouse movements, and even capture or display images on the screen. And it has a network interface that is demonstrably insecure, which can allow an attacker on the network to inject rootkits that completely compromise the PC and can report to the attacker all activities performed on the PC. It is a threat to freedom, security, and privacy that can’t be ignored.

Before version 6.0 (that is, on systems from 2008/2009 and earlier), the ME can be disabled by setting a couple of values in the SPI flash memory. The ME firmware can then be removed entirely from the flash memory space. libreboot does this on the Intel 4 Series systems that it supports, such as the Libreboot X200 and Libreboot T400. ME firmware versions 6.0 and later, which are found on all systems with an Intel Core i3/i5/i7 CPU and a PCH, include “ME Ignition” firmware that performs some hardware initialization and power management. If the ME’s boot ROM does not find in the SPI flash memory an ME firmware manifest with a valid Intel signature, the whole PC will shut down after 30 minutes.

Due to the signature verification, developing free replacement firmware for the ME is basically impossible. The only entity capable of replacing the ME firmware is Intel. As previously stated, the ME firmware includes proprietary code licensed from third parties, so Intel couldn’t release the source code even if they wanted to. And even if they developed completely new ME firmware without third-party proprietary code and released its source code, the ME’s boot ROM would reject any modified firmware that isn’t signed by Intel. Thus, the ME firmware is both hopelessly proprietary and “tivoized”.

In summary, the Intel Management Engine and its applications are a backdoor with total access to and control over the rest of the PC. The ME is a threat to freedom, security, and privacy, and the libreboot project strongly recommends avoiding it entirely. Since recent versions of it can’t be removed, this means avoiding all recent generations of Intel hardware.

More information about the Management Engine can be found on various Web sites, including me.bios.io, unhuffme, coreboot wiki, and Wikipedia. The book ***Platform Embedded Security Technology Revealed*** describes in great detail the ME’s hardware architecture and firmware application modules.

If you’re stuck with the ME (non-libreboot system), you might find this interesting: http://hardenedlinux.org/firmware/2016/11/17/neutralize_ME_firmware_on_sandybridge_and_ivybridge.html

Also see (effort to disable the ME): https://www.coreboot.org/pipermail/coreboot/2016-November/082331.html - look at the whole thread