



**FREE SOFTWARE
FOUNDATION**

A statement by FSF President Richard Stallman on the Intel Management Engine (ME):

"Meltdown and Spectre are errors. Grave errors, to be sure, but not evidently malicious. Everyone makes mistakes. Intel has done far worse with its CPUs than make a mistake. It has built in an intentional back door called the Management Engine. Important as these bugs are, don't let Intel's mistakes distract you from Intel's deliberate attack!"

The Intel Management Engine: an attack on computer users' freedom

By Denis GNUtoo Carikli

With security issues like the Spectre and Meltdown vulnerabilities (<https://www.cnet.com/news/meltdown-spectre-intel-ceo-no-recall-chip-processor/>) discovered in Intel chips in early 2018, it became more important than ever to talk about the necessity of software freedom in these deeply embedded technologies. Serious as though these bugs may be, we cannot let them distract us from the broader issues: Intel considers the Intel Management Engine a feature, while it's nothing more than a threat to user freedom. Thanks to Denis GNUtoo Carikli, we have a new basis for that conversation in this article.

The Intel Management Engine is a tool that ships with Intel chipsets, purportedly to ease the job of system administrators. But in reality, it is another restriction on user freedoms, imposed by a company, and used to control your computing.

Carikli offers a moderately technical explanation of what's happening with Management Engine, the ways in which it restricts rather than empowers users, and how it violates the four freedoms of free software.

Carikli may be best known for his work on the Replicant project (<https://www.replicant.us/>), which he co-founded with Aaron Williamson, Bradley Kuhn, and Grazlano Sorbaioli. He has also worked

on a number of free BIOS/UEFI including coreboot (<https://www.coreboot.org>) and serialICE (https://www.serialice.com/Main_Page).

The Management Engine¹ (frequently abbreviated as ME) is a separate computer within Intel computers, which denies users control by forcing them to run nonfree software that cannot be modified or replaced by anyone but Intel. This is dangerous and unjust. It is a very serious attack on the freedom, privacy, and security of computer users.

The Management Engine started to appear in Intel computers around 2007.²

At first, it was designed to help system administrators and other employees to remotely manage computers³, and was advertised as a computer feature for business customers.

It could, for instance, be used to remotely:

- Power the computers on and off.
- Boot computers from remote storage located on the system administrator's machine or on a server, and take control of the computer that way.⁴
- Retrieve and store various serial numbers that identify the computer hardware.

Over time, Intel imposed the Management Engine on all Intel computers, removed the ability for computer users and manufacturers to disable it, and extended its control over the computer to nearly 100%. It even has access to the main computer's memory.

It now constitutes a separate computing environment that is designed to deny users the control of their computer. It can even run applications that implement Digital Restrictions Management (DRM).⁵ See Defective by Design (<https://www.defectivebydesign.org/>) to learn why DRM is bad.

The remote administration is done through applications running inside the Management Engine, such as AMT (Active Management Technology).⁶ AMT gives remote system administrators the same control they would have if sitting in front of the computer.⁷ AMT can also control Intel Ethernet interfaces and WiFi cards to filter or block network traffic from going in or out of the computer.⁸

Intel has gone as far as to use a free operating system and convert it to nonfree software to attack its users' freedom: The license⁹ of the operating system they use does not give users rights to the source code under a free license, nor does it ensure users' rights to run modified versions of that code on the Management Engine.

We could correct all these problems if the users were able to run fully free software on the Management Engine, or at least, make it not run any code, effectively disabling it. The former is impossible because the Management Engine will only run code that is cryptographically signed by Intel.¹⁰ This means that unless someone finds a flaw in the hardware that enables users to bypass the signature check, users are effectively denied the ability to install the software they wish in the Management Engine.

To prevent free operating systems from being subverted into an instrument that makes attacking users' freedom cheaper and easier, it is important to license their components under the GNU GPLv3 or later whenever possible. This keeps the software free and prevents hardware manufacturers from denying end users the ability to run modified versions of the software. See <https://gnu.org/licenses/license-recommendations.html> to learn about the best licensing strategies to

maximize users' freedom, and in which cases licenses other than the GPLv3 might be suitable.

Despite all Intel's efforts to make the Management Engine inescapable, software developers have had some success with preventing it from loading code. For instance, the Libreboot project disables the Management Engine by removing all the code that the Management Engine is supposed to load on some Thinkpad computers manufactured in 2008, including the R400, T400, T400s, T500, W500, X200, X200s, and X200T.

Also, many Intel computers manufactured in 2006 have the ancestor of the Management Engine which is disabled from the start, such as the Lenovo Thinkpads X60, X60s, X60 Tablet and T60, and many more.

A free software program named intelmetool¹¹ is capable of detecting if the Management Engine is absent or disabled. With more recent hardware, it is not yet possible to fully disable the Management Engine, as some of the hardware needs to be initialized by it. It is however possible to limit the amount of nonfree software running on the Management Engine by removing parts of the code and/or by configuring it to not run some code.¹²

Independently from the Management Engine, other issues affect computers users in very similar ways:

- Many computers use nonfree boot software (like BIOS or UEFI or equivalent) and/or require it to be cryptographically signed by the hardware manufacturer. This raises similar concern for the freedom, privacy, and security of computer users because the boot software is responsible for loading the operating system, and has more control over the computer than the operating system. This issue also affects computers using other architectures such as ARM.¹³
- AMD¹⁴ computers made after 2013 also have a separate computer within the computer, called PSP (Platform Security Processor), which has similar issues.

Because of Intel's attack on users' freedom, to avoid being denied freedom, privacy, and security, computer users wanting to use a machine with an Intel processor must use older computers with no Management Engine, or whose Management Engine is disabled.

Whenever companies follow Intel's path, we will need to design our own hardware to keep being able to escape such attacks on freedom, by ensuring that users can run fully free software on it. This will also create the necessary building blocks that will enable users to benefit from hardware freedoms¹⁵ in the future, when manufacturing technologies are easily available to end users.

For more information on the Intel Management Engine, see:

- Intel & ME, and why we should get rid of ME (<https://www.fsf.org/blogs/licensing/intel-me-and-why-we-should-get-rid-of-me>)
- Active Management Technology": The obscure remote control in some Intel hardware (<https://www.fsf.org/blogs/community/active-management-technology>)

References:

1. Also called SPS (Server Platform Services) on servers and TXE (Trusted Execution Engine) on some mobile or low power devices.
2. For more information about the history of the Management Engine, see pages 27, 28, and 29 of the 2014 book *Platform Embedded Security Technology Revealed*, by Xiaoyu Ruan (ISBN 978-1-4302-6571-9), at Springer (<https://link.springer.com/book/10.1007%2F978-1-4302-6572-6>).
3. The remote management can be done through an application that is running inside the Management Engine. Various applications exist for that, and the best known is called AMT (Active Management Technology).
4. This functionality is part of AMT, and is known as SOL/IDE redirect.
5. For more information about Digital Restrictions Management and the Management Engine, see from page 191 until the end of chapter 8 (Hardware-Based Content Protection Technology) of the book *Platform Embedded Security Technology Revealed*, by Xiaoyu Ruan (ISBN 978-1-4302-6571-9) at Springer (<https://link.springer.com/book/10.1007%2F978-1-4302-6572-6>).

This chapter tries to justify the usage of Digital Restrictions Management (DRM). DRM is totally unacceptable as it requires the users not to be in control of their computers to effectively prevent them from exercising their legal rights (such as fair use, or being able to copy published works). That chapter clearly shows the link between preventing users from controlling their hardware and effective DRM.

FSF's Defective by Design campaign (<https://www.defectivebydesign.org>) has resources to take actions against DRM.

6. AMT is often available on Intel computers designed for business customers, and not on computers designed for consumers. When available, there are often BIOS or UEFI settings to turn it off, but as they are implemented by nonfree software, there is no easy way to know what such settings really do, or to know what the consequences of turning AMT on or off that way really are.
7. This functionality is part of AMT, and is called System Defense. For more detail about that see Intel's System Defense description (https://software.intel.com/sites/manageability/HLAPI_Documentation/Documents/detaileddescription3.htm) and [Intel's documentation on how it works \(https://software.intel.com/sites/manageability/HLAPI_Documentation/Documents/howssystemdefenetworks.htm\)](https://software.intel.com/sites/manageability/HLAPI_Documentation/Documents/howssystemdefenetworks.htm).
8. To do that, Intel used VNC (Virtual Network Computing), a standard protocol to remotely administrate computers, by relaying keyboard, mouse, and display over a network. Many free software programs implementing such protocols can also do that, and can be found in the free software directory (<https://directory.fsf.org>).
9. Here, Intel recently started to use Minix, a free software operating system released under various BSD licenses.

BSD licenses are weak free software licenses that don't prevent software from being used to mistreat users (by removing the freedoms it came with).

Some parts of Minix are released under the Original BSD license, or modified versions of it. This issue makes it impossible to combine such software with software licensed under the GNU GPL licenses. To avoid that issue it is better to choose other weak licenses as explained in this article about the modified BSD license: <https://www.gnu.org/licenses/license-list.html#ModifiedBSD>.

10. This means that, through cryptography, the hardware manufacturer (for instance Intel) decides which code can run on that hardware.

11. See intelmetool (<https://github.com/zamaudio/intelmetool>), a utility for reporting the Management Engine status.

12. This can be done with the me_cleaner program (https://github.com/corna/me_cleaner/). Also, Purism's Librem 13 v2 and Librem 15 v3, sold after 19 October 2017, have already had that done; for machines sold earlier, and the Librem 13 v1, the build_coreboot.sh program (https://code.puri.sm/kakaroto/coreboot-files/raw/master/build_coreboot.sh) can do it. You can also do it with Coreboot 4.6 by enabling the option named "Strip down the Intel ME/TXE firmware." Note that Coreboot is not entirely free software itself. More generally, you can use the me_cleaner program to do this for any Intel computer that has the Management Engine. For more detail about how me_cleaner works see Positive Technologies's article named "Disabling Intel ME 11 via undocumented mode" (<http://blog.ptsecurity.com/2017/08/disabling-intel-me.html>); me_cleaner's documentation on how it works (https://github.com/corna/me_cleaner/wiki/How-does-it-work-%3F); and me_cleaner's documentation on the HAP and the AltMeDisable bits (https://github.com/corna/me_cleaner/wiki/HAP-AltMeDisable-bit).

13. ARM is a computer hardware architecture that is commonly found in small and mobile devices such as smartphones and tablets.

14. AMD is a company that makes computer hardware which is mostly equivalent to Intel hardware, and can replace it, as it can run the same operating systems and applications with no or very few changes.

15. For more details on hardware freedom, see the article on Free Hardware and Free Hardware Designs at <https://www.gnu.org/philosophy/free-hardware-designs.html>.