Proving the Known, EFI/UEFI Exploited for BIOS Level Attacks – Purism

3-4 minutes ⋮ 3/24/2017

- About
- Latest Posts

Todd Weaver

Founder and CEO
PGP Fingerprint: B8CA ACEA D949 30F1 23C4 642C 23CF 2E3D 2545 14F7

We're continuing with a second report (many more coming!) on the "Vault 7" Documents we started digesting recently. There is an extensive section dedicated to EFI/UEFI exploitations. While this threat has been known from a theoretical standpoint from the moment the non-free BIOS replacement–EFI/UEFI–came into existence, the Vault 7 documents published recently now confirm that these threats are real and these weaknesses are actively being exploited.

One interesting read we're focusing on today is the EFI/UEFI "ExitBootServices Hooking" exploit and sample copy-and-paste code to inject a hook into the last execute state of the EFI/UEFI process (the "ExitBootServices").

> Copy-and-paste code was included in the leaks which allow for the exploitation of UEFI-based boot systems by altering the operating system's kernel which is loaded into memory before exiting the UEFI boot sequence. The copy-and-paste code allows for an attacker to insert a custom hook which can be used to arbitrarily alter the operating system's kernel in memory immediately before execution control is handed to the kernel. — Wikipedia's summary.

It is trivial to utilize this exploit:

> Because the ExitBootServices service can be found by getting its pointer from the global EFI_BOOT_SERVICES table, hooking the ExitBootServices call is trivial. […] When you're running in UEFI, that EFI_BOOT_SERVICES table isn't protected by anything, so you can just write directly to it. — Vault 7 ExitBootServices Hooking

The result is that the entire system is compromised. As the page highlights, "At this point, you can do whatever you want."

This type of exploit once-again highlights that security is a game of depth. This exploit is one level below the kernel, which means it has complete control of every level above it, such as the kernel, the entire operating system, any and all applications, network traffic, web application usage, and all user interaction.

The good news is, Purism recently completed the port of coreboot to the Librem 13 v1 (with more ports to come for the rest of our devices), providing a free/libre and open source replacement for EFI/UEFI which avoids all of the exploits mentioned within the documents.

The only long-term approach to protect oneself is to have complete control of the device. Control is the key word, and there is no other way to have complete control than to have as much of the software released under free software licenses where the source code is available to confirm it operates in your best interest and not that of criminals, spies, bad hackers, nations, or thieves.

Confirming that EFI/UEFI has a known and trivial exploit that is built into the standard also confirms that there is no depth too deep to exploit, and the only defense against unwanted stripping of a users' digital rights is to use hardware and software that you control. Purism does just that by releasing all software under a free software license where the source code is available to be audited, reviewed, and scrutinized making a user control their device not the device controlling the user.