

Security Vulnerabilities in UEFI BIOS

August 13, 2024

Jay Talbott
Principal Consulting Engineer
SysPro Consulting

Introduction

A computer system's boot firmware runs at the highest level of privilege and needs to be highly scrutinized for any security vulnerabilities which could be exploited to compromise the performance or behavior of the system. The vast majority of x86/x64-based systems utilize boot firmware, or Basic Input Output System (BIOS) firmware, that is based on the Unified Extensible Firmware Interface (UEFI) specification. Recently several new major security vulnerabilities have been identified with UEFI BIOS implementations. This paper discusses the history of UEFI BIOS, highlights the recently discovered security vulnerabilities, and suggests alternative options for x86/x64 boot firmware to avoid such vulnerabilities.

History of UEFI BIOS

Since the invention of the original IBM PC, all x86/x64-based computer systems have required on-board boot firmware to initialize the hardware and prepare the system to boot an operating system. Early PCs used legacy BIOS firmware, which could boot into DOS, Windows 3.1, and other legacy operating systems. It also provided basic I/O services to the operating system and applications. Modern x86/x64-based systems use BIOS firmware that is compliant with the UEFI specification, which supports booting into modern versions of Microsoft Windows® as well as other operating systems and provides a much broader set of runtime services.

The UEFI specification evolved out of the original Extensible Firmware Interface (EFI) specification¹ that was originally developed by Intel® in the 1990s for the (now discontinued) Intel Itanium® processor. The EFI specification addressed many of the restrictions and limitations of legacy BIOS, for example the lack of support for 64-bit processors and the lack of support for large disk drives. While Intel still owns the original EFI specification, it contributed it to the UEFI Forum², which owns and maintains the current UEFI specification.

Complexity of a UEFI BIOS

A UEFI BIOS is based on a very large and complex code base and requires extensive experience to really know how to work with the code base effectively. Internally a UEFI BIOS is essentially an operating system with a dispatcher that executes various code modules, which are dynamically loaded and run at boot time, rather than being statically linked and simply run in a predetermined sequence.

A UEFI BIOS also provides a broad set of UEFI runtime services, which can be used by the operating system and applications to perform functions and obtain information about the state of the system. However, the only operating system that really uses these UEFI runtime services is Microsoft Windows, so if you are booting an operating system other than Windows, providing support for all of these runtime services in the firmware is completely unnecessary.

Some argue that a UEFI BIOS is far more complicated than it needs to be. For many applications, all that is required is to initialize the CPU and memory, configure the I/O, and boot the operating system, and to do all of that quickly and securely.

Security Vulnerabilities in UEFI BIOS

There has been a long history of security vulnerabilities that have been discovered in UEFI BIOS. While many of them have been fixed along the way, new security vulnerabilities continue to be discovered as

those who seek to find new ways that such systems can be compromised develop more and more sophisticated methods to identify weaknesses that can be exploited. One would think that after more than 20 years fixing the various vulnerabilities that have been discovered that UEFI BIOS would ultimately be reasonably secure, but surprisingly, that's simply not the case.

For example, in just the past year, **four** new major security vulnerabilities have been discovered in UEFI BIOS:

1. "LogoFAIL"³, where malicious code can be embedded into a splash screen logo file that will get executed when that logo file is loaded by a UEFI BIOS to be displayed on the screen.
2. "PixieFail"⁴, where nine new vulnerabilities were found in the network stack that supports booting an operating system from the network.
3. "TPM GPIO fail"⁵, where the unlocked GPIO configuration in virtually all systems allows for compromising the secure boot data stored in the Trusted Platform Module (TPM).
4. "PKfail"⁶, where leaked private keys were used in production by numerous system manufacturers, completely compromising the security of the firmware.

Note that none of these are new vulnerabilities. Rather, they are vulnerabilities that have been there all along, but have just been recently discovered, along with methodology for how to exploit them. And all of them pose extremely serious security risks.

Unfortunately, the discovery of such security vulnerabilities requires firmware updates, which takes time to work their way through the UEFI ecosystem, and which are not always practical to install on systems that have already been deployed. It can take many months, if not years, before fixes are made and updates become available, and many times, even once they are available, they simply don't get installed on the vulnerable systems, leaving them open to attack.

UEFI BIOS Vendor Stance

The major independent UEFI BIOS vendors (IBVs) each gave presentations at the UEFI Fall 2023 Developers Conference and Plugfest that touched on security vulnerabilities. Note that this event occurred prior to when any of the above vulnerabilities were discovered.

American Megatrends (AMI) gave a presentation⁷ on *vulnerability management* that focused on *reactive security*, and pointed out that the rate of UEFI BIOS firmware attacks is increasing.

Insyde Software reported in their presentation⁸ that Insyde has an average of over 20 security incidents *per month* in their UEFI BIOS firmware implementations. The speaker stated that only 50% of such vulnerabilities are fixed within the first 4 months of discovery. He further indicated that the network stack is where many new vulnerabilities have been recently discovered, and the expectation is that there will be many more over the next couple of years. Note that PixieFAIL was discovered after he gave this presentation, consistent with his prediction.

Phoenix Technologies gave a presentation⁹ on *firmware vulnerability scoring*, which focused on rating the severity of each discovered vulnerability.

All of these presentations suggest that when it comes to security vulnerabilities in their respective UEFI BIOS implementations, the various IBVs have a much more *reactive* stance, rather than a *proactive*

stance. They also indicated how slow it can be to develop fixes and issue updates that patch the various security holes, and that in reality not all of the vulnerable systems ever get the updates installed. This just can't continue this way if there's ever hope of having truly secure systems.

Along with the presentations by the IBVs, there was also a presentation by a representative of the Cybersecurity & Infrastructure Security Agency of the United States government¹⁰, who stressed the need for better UEFI security. If he only knew then what was to come in the months that followed that event as the above vulnerabilities were discovered.

Boot Firmware Alternatives

To enable alternative boot firmware solutions, in the early 2010s, Intel Corporation began developing the Intel® Firmware Support Package (FSP)¹¹ for various Intel CPU and SoC families. The FSP for each supported Intel platform includes a binary file that encapsulates all of Intel's proprietary initialization code along with the supporting files needed for integration into a bootloader solution. These FSPs enable third parties to develop alternative boot firmware solutions for Intel processors that are based on open-source projects such as coreboot¹² or Intel's Slim Bootloader¹³.

Similarly, Advanced Micro Devices (AMD) has developed their own packages, such as AMD Generic Encapsulated Software Architecture (AGESA)¹⁴ and AMD Open-Source Silicon Initialization Library (openSIL)¹⁵, which also enable third parties to develop alternative boot firmware solutions for AMD processors that are based on coreboot.

Alternative boot firmware solutions that encapsulate these packages are typically based on smaller and simpler code bases, which are statically linked and have deterministic sequential code execution. They only include what is necessary to initialize the system for the target application. The goal of these boot firmware solutions is to just do the bare minimum required before handing off control to the operating system:

- Initialize the CPU.
- Initialize the memory.
- Configure the I/O.
- Load and handoff control to the operating system.
- Do all of the above quickly and securely.

Because these alternative boot firmware solutions are not based on the same code as a UEFI BIOS, they do not share the same security vulnerabilities that continue to be discovered in UEFI BIOS firmware. For example, the alternative boot firmware solutions that have been developed by SysPro Consulting¹⁶ are not subject to any of the above four major security vulnerabilities that were discovered in UEFI BIOS over the past year.

As another example, Lean Sheng Tan, the head of Open Systems Firmware at 9elements Cyber Security, noted in a recent LinkedIn post¹⁷ that Google Chromebooks, which use an alternative boot firmware solution based on coreboot, have been unaffected by these UEFI BIOS security vulnerabilities.

Conclusion

A UEFI BIOS, which has been the default boot firmware solution for x86/x64-based systems for the past couple of decades, continues to contain numerous security vulnerabilities. This should be alarming to anyone who fully understands the implications and the risks of the recently discovered vulnerabilities. And there appears to be no end in sight to what is yet to be discovered.

While UEFI BIOS is the default solution, there are alternative options for x86/x64 boot firmware solutions which do not share these security vulnerabilities. Companies that develop and manufacture x86/x64-based systems should seriously consider these alternative options in lieu of continuing to use UEFI BIOS as the boot firmware.

About SysPro Consulting

SysPro Consulting is an Intel® Gold partner that specializes in the development of alternative boot firmware solutions for embedded systems that are based on Intel CPUs and SoCs. SysPro spent several years providing engineering consulting services to Intel, working directly with Intel's FSP team. Since that time SysPro has been developing boot firmware solutions for various Intel customers and has been a subcontractor on numerous U.S. government / DoD programs.

Unlike the IBVs, SysPro is a software engineering consulting company. SysPro does not sell *products* nor charge *royalties*. Instead, SysPro sells software engineering services on a purely time and materials basis. SysPro works more like an extension of their clients' existing engineering teams, offloading the boot firmware development from their clients so that they can focus on their core technology. SysPro can customize each boot firmware solution for the specific project requirements and hardware design for the particular application.

At the end of each project, SysPro provides its clients with all of the bootloader source code and supporting documentation so they can then build and support the boot firmware solution themselves. SysPro's clients can then choose to take it from there, or, more commonly, come back to SysPro for any ongoing support, maintenance, or other changes, or for additional boot firmware development projects. SysPro's value is in its expertise and experience in this field, as the SysPro team consists primarily of former Intel employees and others who have been engaged with Intel in one way or another doing boot firmware development.

For more information about SysPro Consulting, reach out to the author of this paper via the contact information below.

Contact information

Jay Talbott
Principal Consulting Engineer
SysPro Consulting
(480) 704-8045
JayTalbott@sysproconsulting.com
<http://www.sysproconsulting.com>

Copyright © 2024 SysPro Consulting, LLC. All rights reserved.

References

- ¹ [Extensible Firmware Interface \(EFI\) specification 1.10](#)
- ² [UEFI Forum](#)
- ³ [The Far-Reaching Consequences of LogoFAIL](#)
- ⁴ [PixieFail: Nine vulnerabilities in Tianocore's EDK II IPv6 network stack.](#)
- ⁵ [TPM GPIO fail: How bad OEM firmware ruins TPM security](#)
- ⁶ [PKfail: Untrusted Platform Keys Undermine Secure Boot on UEFI Ecosystem](#)
- ⁷ [Vulnerability Management in UEFI](#)
- ⁸ [UEFI Goes to Washington](#)
- ⁹ [Call for Collaborative Action: CVSS V4.0 and Firmware Vulnerability Scoring](#)
- ¹⁰ [A Conversation on Bolstering UEFI Cybersecurity](#)
- ¹¹ [Intel Firmware Support Package](#)
- ¹² [Coreboot](#)
- ¹³ [Slim Bootloader](#)
- ¹⁴ [AGESA](#)
- ¹⁵ [openSIL](#)
- ¹⁶ [SysPro Consulting](#)
- ¹⁷ [LinkedIn post by Lean Sheng Tan](#)