THE CHAIN OF TRUST

Keeping Computing Systems More Secure
Revised: August 2019

Authors:

Richard Wilkins, Ph.D.
Phoenix Technologies, Ltd.
Dick_Wilkins@phoenix.com

Toby Nixon, Senior Standards Program Manager
Windows and Devices Group, Microsoft Corporation
Toby.Nixon@microsoft.com

June 2016

Understanding the Chain of Trust and Its Vital Role in Keeping Computing Systems Secure

Imagine a multi-million-dollar mansion with a sturdy fence surrounding the grounds, full-time security guards, cameras and alarms systems. Seems secure doesn't it? But now imagine that the same mansion has a hidden underground tunnel and passageway that the mansion's owner and the security team know nothing about. In this scenario, all the above-ground security measures are easily defeated if a burglar discovers the hidden tunnel and a way into the mansion.

Until recently, a similar vulnerability existed in the computing world, and in many cases, still does. The "tunnel" into computing systems is through unsecured firmware that runs when computing devices start up, completely bypassing perimeter defenses like firewalls and anti-virus software. Once in, hackers can gain near unrestricted access to target systems for profit or mischief.

Unlike the unaware mansion owner, the computing industry is keenly aware of the firmware "tunnel" and some are determined to shut it down. As will be detailed in a series of white papers on the topic starting with this document, the UEFI (Unified Extensible Firmware Interface) Forum is leading the charge to lock down firmware using an approach called Chain of Trust (CoT). When the CoT is implemented as recommended by the UEFI Forum, users gain more confidence that their computing system is as safe and free from different classes of attacks as possible.

When it comes to security measures, there's always a risk that the cure is worse than the disease. In other words: Is ensuring security so difficult that it's not worth the effort and will it impact usability and flexibility? Fortunately, the CoT mechanisms are straightforward to implement and well-documented by the UEFI Forum with free instructions and tools on tap to assist firmware developers as needed. Moreover, secured systems suffer no loss of performance and offer the same levels of freedom and flexibility as unsecured systems. As such there is no valid reason for modern computing systems—everything from servers to laptops to mobile phones to Internet of Things devices—not to be fully secured using the CoT approach.

The Threat to Firmware

Regardless of the manufacturer or application, computing systems are all comprised of a series of layers starting with the raw hardware or silicon chips that move electrons around and process the 1s and 0s on up to the flashy operating system and all the cool applications we all know and love. In between are layers of specialized executable code known as firmware responsible for translating requests and actions from the operating system into something the hardware can use in order to do work for the user.

In the pre-Internet days, not much consideration was given to computer security. But as more and more computer systems gained connectivity, hackers soon discovered operating system flaws and other vulnerabilities that would give them access to millions of systems. This led to the creation of an entire industry devoted to computer security. And those efforts have worked, making it increasingly challenging for hackers to have much impact.

Seeking an easier route, malware developers turned their attention to the firmware that runs before the operating system, leading to the creation of attack codes known as rootkits and bootkits. Rootkits and bootkits are entities that aren’t part of the original firmware, but insert themselves into the firmware layer. They blend in making them virtually untraceable. A firmware rootkit for example, has to go through “kernel-land” to reach the firmware layer – just like how bad guys would have to go through the village and various check points to enter the mansion. Obtaining full access to the hardware, the malware can do nearly anything it wants from erasing hard drives to logging keystrokes and quietly sending information to far-off destinations. Since the operating system assumes the firmware is trustworthy, such attacks avoid detection and even scrubbing the hard drive may not create a secure system.

In response to these potential malware threats, a number of specifications and hardware tools were developed to provide some protection during the pre-boot process. For instance, the US National Institute of Standards and Technology released a number of guideline documents related to firmware security during pre-boot, including NIST 800-147, which outlines a secure firmware update mechanism, and NIST 800-155, which outlines a “measured boot” that attempts to detect and report unauthorized modifications to firmware or systems configuration.

The NIST guidelines provide a framework designed to keep the firmware from being improperly modified. While a step forward, these guidelines alone are not enough to prevent a system from starting up in a compromised state. To help prevent this, a mechanism needs to be in place to validate every section of code that loads during the pre-boot process to ensure it is safe and unmodified. That mechanism is UEFI Secure Boot.

From Root of Trust to Chain of Trust

When a CPU starts up, it executes only a few very specific instructions at a very specific address. Nothing is initialized yet. There are simply not enough resources to start any code. At this point, all the system can do is to find, validate, install and run a small initial piece of firmware. In the UEFI Secure Boot process this SEC (Security) Phase, as initial turn-on is called, forms the basis for the Root of Trust. Changing what happens during the SEC Phase, while theoretically possible, would be very difficult and would involve gaining physical access to the system and modifying or replacing hardware. CPU vendors have now started checking "signatures" of this initial piece of firmware to ensure it has not been modified improperly and make sure the chain of trust has a solid start.

From this very secure basis flows the Chain of Trust used in UEFI Secure Boot. The trust is maintained via public key cryptography. Hardware manufacturers put what’s known as a Platform Key (PK) into the firmware, representing the Root of Trust. The trust relationship with operating system vendors and others is documented by signing their keys with the Platform Key.

Security is established by requiring that no code will be executed by firmware unless it has been signed by a “trusted” key whether it’s an operating system boot loader, a driver located in the flash memory of a PCI Express card or on disk, or an update of the firmware itself. Key Exchange Keys (KEK) can be added to the UEFI key database so trusted third-party applications can use other certificates as long as they are signed with the private part of the PK.

To manage the signing process, a centralized Certificate Authority (CA) is used, currently operated by Microsoft and open to everyone in the industry. The opportunity exists for other organizations to set up a certificate authority on a broader basis or for their exclusive use as well. The Microsoft-operated UEFI CA, which is available for a nominal fee, has been accepted broadly across the industry, including by most major Linux distributions and third-party developers.

The Secure Boot Mandate

While most developers see the need for securing firmware using a CoT mechanism, some believe it may be too big, complex and slow down their systems. What's worse, some truly believe that their system environment is so constrained that it is impossible to penetrate. Unfortunately, they are likely wrong.

Providing users with confidence that their applications are operating in a secure environment is a fundamental and reasonable expectation to be met by the computing industry. Getting everyone on board with it can be met through industry standards combined with making true CoT easy for all developers to implement.

Currently, the developers of PCs, tablets and some larger server systems have been the first to embrace total CoT. The hope for the future is to get this process into the ever-growing branches on the tree of computerized devices: smart phones, wearable computers, computerized traffic controls, airport flight monitoring, and definitely into the huge area of medical monitoring and automated drug delivery devices where security of the device operation is absolutely critical.

Computer engineers, developers, and influencers in the companies that are building, or will be building, new computing, Internet, and autonomous devices should adopt CoT technology to ensure security. As the world adds ever more "smart" devices to lives worldwide, the threat of hacking grows right along with the technology. Computerized systems, to be secure today, should implement CoT.

Both management and engineering need to promote the use of CoT. A company, well-versed on CoT, realizes that this is not hard to implement and the benefits cannot be overlooked, or the risk of inaction understated. The standards already exist. They do not have to develop anything from scratch. There is a good solid base to work from and they only need to implement it in their products.

Tapping the Latest Technology

Anyone can access UEFI Forum CoT specifications and easily understand the technology, Forum member or not. UEFI specifications support multiple platforms and architectures. What's more, UEFI specifications are designed to promote cross-functionality to support adoption across multiple operating systems.

The UEFI Forum offers two levels of membership to any interested party: Contributor and Adopter. By joining the UEFI Forum, your company will have the opportunity to contribute to the evolution of the UEFI specification, in addition to other membership benefits. The Adopter membership is complimentary and allows for free access to UEFI technical tools and design guides. For a fee, the Contributor memberships allows for participation at a deeper level. For more details on membership options, please visit: http://uefi.org/join.

About UEFI Forum

The UEFI Forum, a non-profit industry standards body, champions firmware innovation through industry collaboration and the advocacy of a standardized interface that simplifies and secures platform initialization and firmware bootstrap operations. Both developed and supported by representatives from more than 300 industry-leading technology companies, UEFI Forum specifications promote business and technological efficiency, improve performance and security, facilitate interoperability between devices, platforms and systems, and comply with next-generation technologies.

The Forum's spheres of input and influence are large: Membership represents major voices from all players in the industry—open source to proprietary technology, hardware to software, mobile to stationary devices. The Forum collaborates with other standards groups that are essential to computing. For more information about the UEFI Forum and current specification go to www.uefi.org.

# # #