Leverage Modern Hardware Security Features

Trouble beneath the surface

An emerging class of threats seeks to exploit vulnerabilities and defeat security mechanisms below the software layer. Malicious actors seek to gain persistence in machine firmware and bypass software security boundaries. However, the latest systems are equipped with countermeasures and mitigations that address these threats. These countermeasures include modern interfaces for managing firmware, support for verifying integrity at boot time, hardware virtualization features, and anti-exploitation features. In order to realize these benefits, enterprises must tighten hardware refresh cycles and enable firmware security features when provisioning new systems. This includes traditional desktops, servers, and laptops as well as smartphones and tablets in the rapidly growing mobile ecosystem.

At a minimum, workstations should be refreshed every 3-4 years and servers every 5-7 years to ensure support. One study indicates systems more than 4 years old are up to 3 times more expensive to maintain [1], while another indicated benefits of system replacement can be seen after as little as 2 years [2]. Mobile devices are often unsupported and obsolete after 3 years, making their prompt replacement critical.

Unified Extensible Firmware Interface (UEFI)

UEFI defines an industry-standard firmware environment common across many different makes and models of computing devices. System vendors are moving toward a UEFI-focused future to accelerate the adoption of new technologies and the rate at which updates are delivered. Most traditional computing platforms have supported UEFI since 2010, and industry plans to phase out support for legacy Basic Input/Output System (BIOS) and Compatibility Support Module (CSM) by 2020 [3]. Some products will not be updated to support UEFI and these older operating systems and hardware components will fail on newer systems. Enterprises will need to update in order to adapt. Figure 1 shows improvements provided by UEFI versus legacy mode.

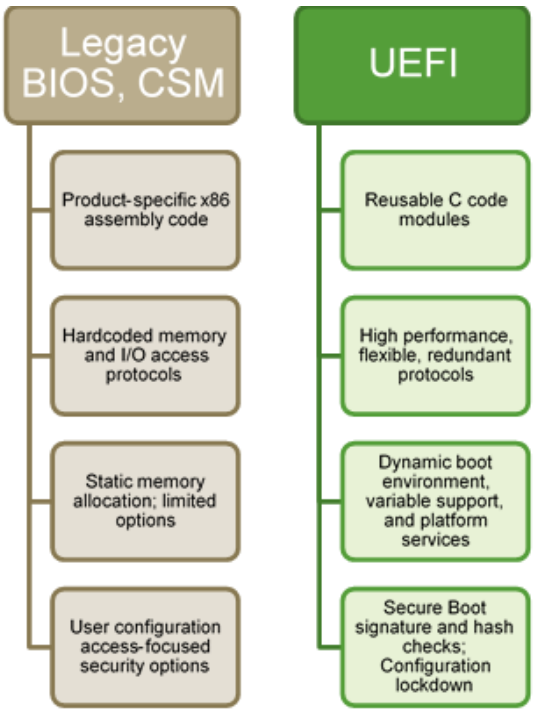

Figure 1: Feature comparison between boot mechanisms.

Products that before saw only bi-annual BIOS updates are now updated with new UEFI firmware on a quarterly or monthly basis to rapidly counter the latest vulnerabilities. Enterprises should adopt UEFI native boot to ensure timely updates and added integrity checks for early-boot firmware.

UEFI Secure Boot

Secure Boot provides a mechanism to counter boot-time malware threats by cryptographically verifying firmware, kernels, and drivers. Enterprises should enable Secure Boot and utilize OS products, hypervisors, and system components that support Secure Boot. Most UEFI implementations since 2012 support Secure Boot. Each executable boot binary, including firmware and software, is checked against a blacklist database and multiple trusted databases. Secure Boot can prevent the use of unapproved OS images, component controllers, boot methods, boot-time malware (such as LoJax), rootkits, outdated kernels, obsolete or evil drivers, and more. OS kernels can extend Secure Boot validation into the software environment by checking signed drivers and executables with root privileges, thus blocking off-the-shelf exploit tools like Mimikatz and Metasploit.

Most systems ship with Secure Boot in standard mode, preconfigured to support Microsoft Windows® and Linux® distributions such as Red Hat Enterprise Linux®. Secure Boot can optionally be configured in custom mode for enterprises

with additional security needs. Custom mode allows infrastructures to trust in-house signatures and hashes to change the scope of permissible boot content. For example, standard mode would have initially trusted malware Superfish while custom mode could have blocked Superfish from day one.

UEFI Lockdown Recommendations

Beyond Secure Boot, UEFI configuration provides the opportunity to mitigate misconfiguration and counter insider threats [4]. Many business class systems provide fully interactive configuration interfaces with options to tune a wide variety of system parameters. Servers may additionally provide the option of remote configuration management. Factory default settings usually focus on compatibility over system lockdown and boot hardening. Enterprises should determine a secure baseline of settings to apply to each endpoint.

To prevent abuse or misconfiguration of firmware features, set a UEFI administrator password. Disable unused device components to prevent data loss through cameras, wireless, and external data ports. Restrict boot order to only devices necessary for OS boot to protect against physical tampering and misconfiguration. Enable platform hardening features provided by hardware and firmware vendors to enforce firmware integrity. Disable legacy settings and components where possible to limit weaknesses in the boot process.

Trusted Platform Module (TPM)

TPM functions as a system integrity observer and trust anchor. Many OS distributions and hypervisor solutions leverage TPM for credential storage and system integrity features that are transparent to enterprise users. Most UEFI implementations can store boot-time integrity measurements in the TPM, enabling their safe storage. TPMs are also equipped with random number generators, secure memory, and cryptographic key generation algorithms that may be compliant with FIPS 140-2 [5]. TPMs come in two variants -- discrete TPM chips affixed to motherboards, or firmware TPM (fTPM) implementations provided by processor manufacturers such as Intel® PTT and AMD® fTPM.

Enterprises should enable and activate a TPM when present to utilize its features, and upgrade to systems with TPM 2.0 whenever possible. Windows 7® and newer automatically leverage TPM for BitLocker® – a data-at-rest solution that leverages encryption and TPM state. Additionally, Windows 10® leverages TPM for Credential Guard – a runtime vault with TPM key-brokered access – and System Guard – a system integrity attestation solution that aims to identify changes to previously known-good software at runtime. Red Hat Enterprise Linux® (RHEL) can be configured to leverage TPM for kernel monitoring via Integrity Measurement Architecture (IMA) and data-at-rest drive encryption through Linux Unified Key Setup (LUKS).

Hardware Virtualization

Modern operating systems rely on virtualization features previously limited to use by type-1 hypervisors like VMware® and Xen®. Virtualization extensions allow operating systems to separate processor cores, regions of memory, and I/O devices based on privilege level and the individual application in use. Control flow integrity and memory tagging allow kernel-level functions to be secured from user-level functions via hardware mechanisms – an improvement over older, purely software-based solutions. Microsoft Windows Defender® Application Guard (WDAG), Microsoft Virtualization-Based Security (VBS) including Hypervisor-Enforced Code Integrity (HVCI), and Linux Containers (LXC) all lose functionality when utilized on older platforms lacking hardware virtualization support. Enterprises should acquire systems with hardware virtualization support and enable use of virtualization features in UEFI configuration to gain the most benefit.

Mobile security is a growing concern

Mobile devices make up an increasing share of the modern computing landscape. Hardware solutions that initially focused on low power consumption have evolved into products that rival the performance and versatility of modern laptops. Malicious actors increasingly target mobile devices which has led to hardware-backed improvements previously limited to traditional platforms. Intense competition from multiple strong hardware development companies has resulted in significant innovation.

ARM® Trust Zone® and Trusted Execution Environment (TEE)

Trust Zone segregates System on a Chip (SoC) processor hardware into a secure world and an unsecure world. The secure portion is tasked with processing sensitive data such as biometrics and credentials. The unsecure portion handles user apps and services [6]. Each vendor has a unique implementation of Trust Zone. Some vendors physically separate hardware components and traces while others dynamically tag and flex the assignment of SoC processor cores based on which security level needs more performance at a given time. Look for features like the Apple® Secure Enclave, Google® TitanM, and Samsung® TEEGRIS and KNOX solutions.

Pointer Authentication Codes (PACs)

Version 8.3-A of the ARM instruction set introduced an ability to place authentication codes within memory addresses used to store pointers. Malicious actors typically rely upon buffer overflows, memory leaks, and other memory attacks to change the target of pointers and cause jumps to malicious programming (commonly referred to as Return Oriented Programming exploits). Established solutions, such as Address Space Layout Randomization (ASLR), are routinely defeated by malicious actors [7]. PACs enable the hardware to check whether a pointer has been maliciously altered since originally written into memory. This provides modern mobile systems with a new defense that significantly raises the complexity of exploitation.

New threats, modern solutions

Users and OS environments are not the only things under threat anymore. Firmware, expansion devices, and sensitive memory locations also need protection from attack. Modern systems offer hardware-backed security mechanisms to combat the latest threats, but these are only effective if modern hardware is deployed throughout an enterprise and refreshed on a regular basis.

Works Cited

[1] "True cost of not replacing computers revealed in Microsoft study: more than $4,000 each." Microsoft, 2018. Retrieved from: https://news.microsoft.com/en-nz/2018/10/16/true-cost-of-not-replacing-computers-revealed-in-microsoft-study-more-than-4000-each

[2] "Replacing Enterprise PCs: The Fallacy of the 3-4 Year Upgrade Cycle." J. Gold Associates (2014). Retrieved from: https://i.crn.com/sites/default/files/ckfinderimages/userfiles/images/crn/custom/INTELBCCSITENEW/WhitePaper_EnterpriseRefresh.pdf

[3] "The PC BIOS will be killed off by 2020 as Intel plans move to pure UEFI." Ars Technica, 2017. Retrieved from: https://arstechnica.com/gadgets/2017/11/intel-to-kill-off-the-last-vestiges-of-the-ancient-pc-bios-by-2020

[4] "UEFI Defensive Practices Guidance." NSA, 2017. Retrieved from: https://www.nsa.gov/Portals/70/documents/what-we-do/cybersecurity/professional-resources/ctr-uefi-defensive-practices-guidance.pdf

[5] "Security Requirements for Cryptographic Modules." NIST, 2001. Retrieved from: https://csrc.nist.gov/publications/detail/fips/140/2/final

[6] Arm (2014. A technical on TEE and ARM TrustZone. Retrieved from https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/a-technical-report-on-tee-and-arm-trustzone

[7] Google Project Zero Team (2019). Examining Pointer Authentication on the iPhone XS. Retrieved from https://googleprojectzero.blogspot.com/2019/02/examining-pointer-authentication-on.html

Disclaimer of Endorsement

Contact

Client Requirements / General Cybersecurity Inquiries: Cybersecurity Requirements Center, 410-854-4200, Cybersecurity_Requests@nsa.gov
Media inquiries / Press Desk: 443-634-0721, MediaRelations@nsa.gov