Serianu Cyber Security Advisory

UEFI Secure Boot Customization

Serianu SOC Advisory Number:

TA – 2020/0016

Date(s) issued:

13th October 2020

OVERVIEW

UEFI is a replacement for the legacy Basic Input Output System (BIOS) boot mechanism. It provides an environment common to different computing architectures and platforms, configuration options, improved performance, enhanced interfaces, security measures to combat persistent firmware threats and support for a wider variety of devices and form factors.

Secure Boot customization enables administrators to realize the benefits of boot malware defenses, insider threat mitigations and data-at-rest protections.

This advisory provides a detailed recommendation for customizing Secure Boot which provides a validation mechanism that reduces the risk of successful firmware exploitation, a means attacker can use to gain persistent access to victim networks.

1. Unified Extensible Firmware Interface (UEFI)

Unified Extensible Firmware Interface (UEFI) is an interface that exists between platform hardware and software. It is defined and updated via specifications maintained by the UEFI Forum industry body. UEFI defines a consistent Application Programming Interface (API) and a set of environment variables common to all UEFI platforms.

2. UEFI Secure Boot

Secure Boot is a feature added to UEFI specification where each binary (module, driver, kernel, etc.) used during boot must be validated before execution. Validation involves checking for the presence of a signature that can be validated by a certificate or by computing a SHA-256 hash that matches a trusted hash.

3. Platform-Specific Caveats

The extent to which Secure Boot validates the boot process varies based on platform and boot configuration. In general, most enterprise UEFI implementations provide the following options:

- **Thorough or Full Boot** provides the maximum amount of protection by using Secure Boot throughout the boot process. Integrity, signature and hash checks are performed. All authorized firmware binaries are executed.
- **Fast Boot or Minimal Boot** minimizes boot time by skipping numerous checks, which may or may not include Secure Boot checks. Boot speed is prioritized over some security features and/or additional features and peripheral support at boot time.
- **Automatic Boot** attempts to detect when changes have occurred to the early stages of UEFI boot. Automatic Boot invokes.
- Always prefer the thorough or full boot option when unsure of the vendor implementation.

4. Use Cases for Secure Boot

- **Anti-Malware:** Secure Boot shares similarities with allow listing technologies. Rather than looking for malware via a long deny list of known-bad signatures, Secure Boot works from a short allow list of trusted certificates and hashes. Any binary that fails validation is prevented from running at boot-time.
- **Insider Threat Mitigation:** Organizations may block access to USB ports, restrict network use, and monitor user activity to combat insider threats. Secure Boot can help by closing a threat vector many organizations may not plan for. Few restrictions and monitoring capabilities can cope with an insider that has physical access to a machine.
- **Data-at-Rest:** Secure Boot can interact with Microsoft BitLocker and Linux Unified Key Subsystem (LUKS) Full Disk Encryption (FDE) solutions. Secure Boot configuration data is recorded to the TPM at boot time. BitLocker and LUKS (via extension) can use the TPM when wrapping keys for storage.

Recommendations

For system administrators and infrastructure owners:

- Machines running legacy BIOS or Compatibility Support Module (CSM) should be migrated to UEFI native mode.
- Secure Boot should be enabled on all endpoints and configured to audit firmware modules, expansion devices, and bootable OS images (sometimes referred to as Thorough Mode).
- Secure Boot should be customized, to meet the needs of organizations and their supporting hardware and software.
- Firmware should be secured using a set of administrator passwords appropriate for a device's capabilities and use case.
- Firmware should be updated regularly and treated as importantly as operating system and application updates.
- A Trusted Platform Module (TPM) should be leveraged to check the integrity of firmware and the Secure Boot configuration.

Information Sharing

We encourage any organization or individual that has access to secure boot customization to share it with us through our email info@serianu.com.