Electronic Design

NILESH BADODEKAR, Principal Applications Engineer, Memory Solutions, Infineon Technologies, www.infineon.com.

Secure Tomorrow's Data Centers with Platform Firmware Resiliency

Security of data centers becomes increasingly vital as cyberthreats attack one of their most critical layers: platform firmware.

As the digital era evolves and proliferation of AI increasingly depends on data centers, the security of these systems becomes ever-more critical. Cyberthreats that target one of the most important layers — the platform firmware — are becoming more frequent and sophisticated. The integrity and authenticity of this firmware is vital, as firmware manages core functions, including system initialization, hardware configuration, and low-level operations, all of which are essential for secure and reliable system performance.

In response to these challenges, the National Institute of Standards and Technology (NIST) introduced the SP800-193 standard, a framework for achieving platform firmware resiliency (PFR). This article explores the main components of PFR, clarifies the differences between "secured boot" and "measured boot," outlines the key stages of a server's boot process, and highlights the significance of secured memory in implementing the guidelines outlined in NIST SP800-193.

NIST SP800-193: The Standard for Platform Firmware Resiliency

The NIST SP800-193 standard provides guidelines for securing firmware, which is often targeted by attackers seeking low-level entry points into a system. The standard revolves around three principles:

1. **Protection:** Safeguarding platform firmware, including UEFI BIOS and BMC firmware, from any form of unauthorized modifications or tampering. This relies on robust mechanisms such as cryptographic validation and secured firmware storage to protect the firmware's integrity.

2. **Detection:** Involves identifying and alerting upon unauthorized changes or attempted corruption of firmware. Detection tools such as cryptographic hash verification provide real-time anomaly identification.

3. **Recovery:** Enables the restoration of secured and, therefore, reliable functionality following an attack. The key capability here is seamlessly rolling back from the active firmware to a trusted, validated "golden" image.

Together, these principles enable systems to minimize the risks related to firmware-based attacks while maintaining continuous operations for critical infrastructure

Secured Boot vs. Measured Boot

A critical aspect of implementing PFR is ensuring that threats to system integrity are mitigated during the boot process. This requires leveraging techniques like secured boot and measured boot.

Secured Boot: Checking Code Trust Before Execution

Secured boot is designed to allow only authenticated and trusted firmware to execute when the system boots up. At the start of the boot process, the hardware root of trust (HRoT) is initiated to confirm the integrity of the first firmware component, typically the UEFI or BIOS image, using cryptographic digital signatures. Each subsequent component in the boot chain is then validated in a secure, sequential manner (e.g., OS loaders and kernel).

If a single component fails these checks, the system halts the boot process to prevent compromised firmware and software from being loaded. In such instances, recovery mechanisms are initiated, which can replace the compromised firmware with a validated "golden" image stored in secured memory. Secured boot, therefore, is a *proactive* defense mechanism that aims to prevent the execution of tampered firmware.

Measured Boot: Recording Firmware Integrity

Measured boot, in contrast, doesn't prevent the execution of firmware. Rather, it tracks its state for auditing and anom-

aly detection. As firmware executes (e.g., UEFI or OS components), cryptographic hash values of these components are computed and saved to platform configuration registers (PCRs) within the Trusted Platform Module (TPM).

This series of cryptographic measurements enables the generation of an attestation report, which records the history and integrity of the boot sequence. This recorded data can then be validated by a remote attestation server or system administrator to determine whether any unauthorized modifications occurred along the way.

Measured boot is a *reactive* mechanism, focusing primarily on providing visibility into the firmware execution process and enabling corrective actions based on anomalies.

The Synergy Between Secured Boot and Measured Boot

By combining secured boot and measured boot techniques, system designers can establish a multi-layered security approach. The two mechanisms complement each other: Secured boot actively prevents threats by blocking unauthorized code from being executed and **avoids** code replacement, while measured boot **detects** code replacement and provides visibility and accountability, allowing administrators to validate the system's boot health in real-time.

Measured boot extends the scope of integrity verification beyond what secure boot offers, providing enhanced visibility and detection capabilities, especially in scenarios where data-driven attacks (like counterfeit configuration files) can compromise system integrity. Together, these mechanisms create a robust foundation for system security. This holistic approach is a key requirement for achieving NIST SP800-193 compliance.

The Server Boot Process: A Temporal Overview

The boot process for server platforms involves several critical steps, forming a precise sequence that must be secured to maintain a stable and trustworthy system. Below is a step-by-step outline:

1. Power-on and HRoT initialization:

The first step begins when the server is powered on (t=0). At this point, the HRoT initializes and validates the initial firmware, such as BIOS or BMC, through signature verification.

2. BIOS/UEFI initialization:

After HRoT validation, the UEFI firmware loads and continues the secured boot process. This involves validating additional components in the boot chain, such as the OS bootloader and associated configurations, taking care that all code executed is cryptographically signed and verified.

3. Hash generation and TPM logs:

Hash values for firmware, the OS bootloader, critical drivers, and configurations are computed and sequentially recorded in TPM PCRs. This allows for a verifiable log of the entire boot sequence to be maintained.

4. System integrity verification and alerting:

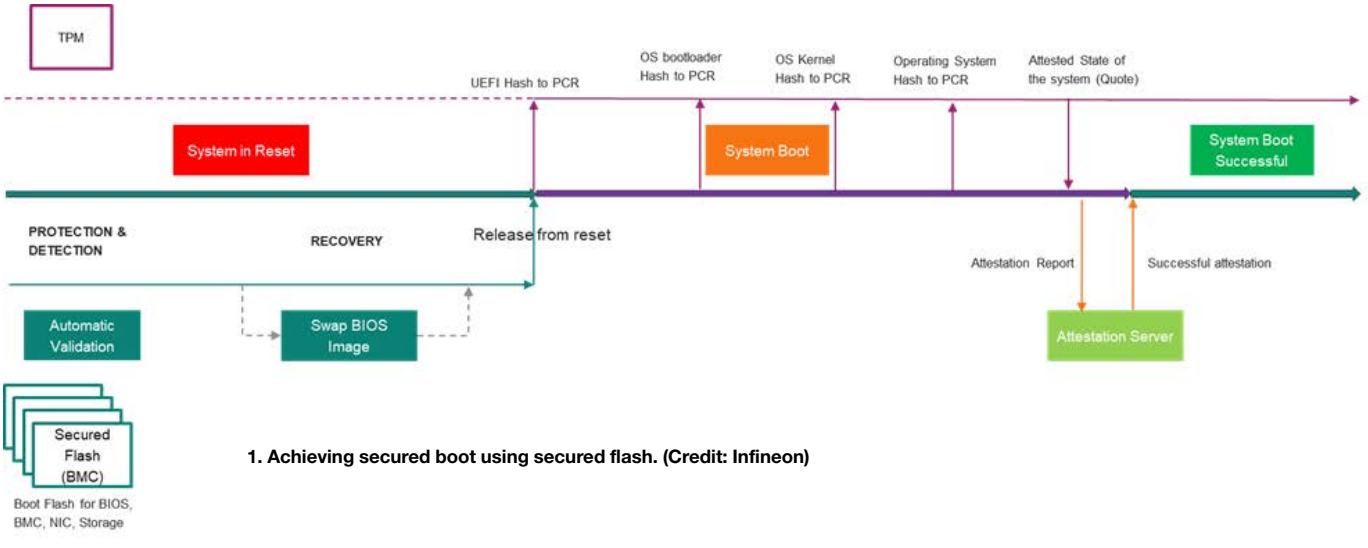
Anomalies in the boot process are identified through verification of the computed hash values against reference hashes. If an issue is discovered, the administrator receives an alert, and the system halts further execution.

5. Recovery options:

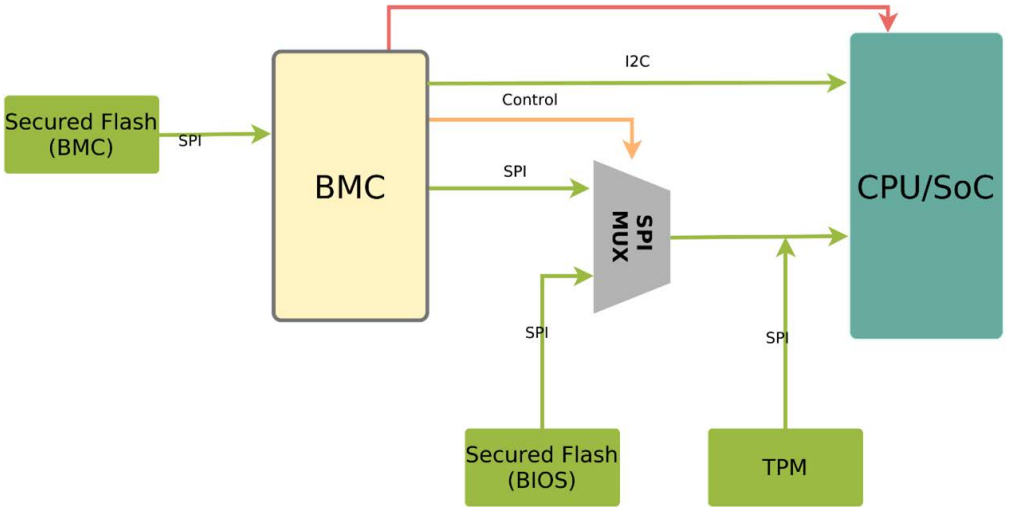
If tampering or corruption is detected during the earlier boot stages, the system provides automatic recovery via secured memory. During this process, the compromised firmware is replaced with a clean version from a protected, "golden" backup image, and the system reboots securely.

6. OS boot and continuous security:

Once the firmware and critical system components are authenticated by the HRoT and validated through the TPM logs, the server transitions to load the operating system, tak-



1. Achieving secured boot using secured flash. (Credit: Infineon)



2. Achieving platform firmware resiliency using secured NOR flash. (Credit: Infineon)

ing care to begin from a secured base.

The Role of Secured NOR Flash Memory in Achieving PFR

Secured memory *(Fig. 1)* can be an essential component in addressing the three pillars of NIST's PFR requirements.

Protection

Secured memory integrates hardware-based roots of trust, preventing unauthorized modifications to stored firmware. This includes cryptographic authentication for each read operation, which prevents TOCTOU (Time-of-Check Time-of-Use) attacks by validating all firmware data in real-time.

Detection

During the boot process, secured memory supports measured boot by performing on-the-fly cryptographic hashing and verification. This checks the integrity of firmware components and enables anomaly detection through TPM-backed attestation logs.

Recovery

Secured memory facilitates recovery by maintaining a validated "golden" firmware image in a secured partition. If corruption is detected, the system can seamlessly switch to this pristine copy to restore operations without requiring further intervention.

This integrated and hardware-enforced approach eliminates the need for additional expensive components, such as FPGAs, while still adhering to stringent security requirements.

Conclusion

Securing datacenters of the future relies on adopting robust security frameworks such as the one specified by the NIST SP800-193 standard. By implementing PFR through secured boot, measured boot, and secured memory technology, organizations can proactively protect platform firmware, detect anomalies, and recover from tampering *(Fig. 2)*. Secured memory plays a pivotal role in achieving these objectives by providing hardware-enabled roots of trust, seamless firmware validation and recovery, and audit capabilities.

A combined approach of secured and measured boot strategies, along with advanced secured memory solutions, is key to addressing the challenges faced by modern data centers. Together, they form a reliable and cost-effective solution that safeguards critical infrastructure, allowing uninterrupted operations and mitigating current and future security risks.

The combined solution of Infineon's SEMPER Secure NOR Flash and InsydeH2O UEFI BIOS, along with Supervyse OPF OpenBMC firmware, addresses the challenges of evolving security threats and meets the requirements laid out by the NIST SP800-193 standard. The SEMPER Secure NOR Flash provides robust hardware-based protection for end-to-end firmware integrity, real-time validation, and streamlined recovery functionality. And the InsydeH2O UEFI BIOS enhances security with trusted boot, secured boot, and measured boot capabilities, providing firmware integrity and proactive defenses.

Nilesh Badodekar is a Principal Applications Engineer at Infineon Technologies. He oversees engineering projects with a variety of Infineon's memory products, ranging from NOR flash to SRAMs and FRAMs, with customers and partners. Nilesh has over 16 years of experience in memory applications in the automotive and industrial segments, having worked in various capacities. He holds a Master of Technology in Embedded Systems from the Indian Institute of Technology, Kharagpur, India.