DEFENDING THE FOUNDATION
OF THE ENTERPRISE
ECLYPSIUM.COM | @ECLYPSIUM

Firmware is Critical Software

Position Paper Submitted by John Loucaides, VP Federal Technology, Eclypsium

Summary: **To ensure the security of the software supply chain, it is recommended that NIST specifically designate firmware as critical software.** Firmware is software that is just stored differently. It forms the foundational layer of software, and as such, performs functions that are essential to correct operation of devices. NIST SP 800-53 Rev. 5 repeatedly underscores the importance of applying the same security protections to firmware as are required for other types of software. However most organizations lack visibility into firmware, and basic security precautions are often ignored.

Firmware and the Criteria for Critical Software

Firmware is pervasive in every computing device. Typical IT equipment (laptops, desktops, servers) has more than a dozen internal components such as UEFI/BIOS system firmware, Trusted Platform Module (TPM), peripheral devices, storage devices, or network interface cards. Each component runs millions of lines of code, developed by a myriad of vendors in a complex supply chain. For network equipment, IoT, OT, and other appliances, firmware goes beyond the component-level software to also include all software, the operating system and applications that are critical to the appliance yet invisible to users and administrators. The firmware in these devices operates at the **highest level of privilege** (sub-ring-zero), because it has **direct access to critical resources** and acts as the **foundational dependency** for the OS and application software (hardware initialization)**,** and is **fundamental to trust** (hardware root of trust and OS boot) in a device. Threat actors increasingly target firmware because a successful attack is stealthy, persistent, and provides ultimate control over the device itself. Finally, firmware risks are those that stand to impact mission readiness the most. The **potential for harm** encompasses initial access, loss or modification of data, persistence, bypassing controls, and even the **destruction of the computing device itself**. Therefore, firmware clearly meets the definition of critical software in the Executive Order:

> *"The security and integrity of "critical software" — software that performs functions critical to trust (such as affording or requiring elevated system privileges or direct access to networking and computing resources) — is a particular concern. Accordingly, the Federal Government must take action to rapidly improve the security and integrity of the software supply chain, with a priority on addressing critical software." Executive Order*

Attacks On Firmware Are Increasing

CISA has issued repeated warnings about threat actors targeting firmware, including raising firmware security as a critical issue in a recent RSA presentation. In April 2021, the NSA, CISA, and the FBI issued a joint report highlighting five CVE's used by the Russian SVR during recent supply chain attacks. Similarly, the Chinese APT5 group was attributed as the source of attacks on a Pulse VPN zero-day vulnerability. A new TrickBot ransomware module targeting UEFI firmware was identified in December.

Microsoft recently issued a report indicating that "83% of businesses have experienced a firmware attack in the past two years," and the "National Vulnerability Database (NVD) has shown more than a five-fold increase in attacks against firmware in the last four years, and attackers have used this time to further refine their techniques and get ahead of software-only protections."

Despite the importance of firmware, most organizations lack visibility into their firmware attack surface. While firmware is just another form of software, tight control by manufacturers and transparent execution before or in parallel with other software (e.g. operating systems and applications) can make it "invisible." As a result, few organizations are able to create a comprehensive inventory of their firmware, determine who manufactured it or what version they have, and whether it is up-to-date. Most organizations patch their operating systems and application software regularly, but firmware patching often lags behind or is not done at all. Like other software, though, firmware is subject to a complex supply chain of shared code, allowing a single error or vulnerability to affect many devices from different manufacturers, just as Eclypsium research into Baseboard Management Controller (BMC) firmware has shown. **If firmware is not explicitly labeled as critical software, there is a real danger that organizations will fail to protect this critical attack surface.**

Implications

Calling out firmware as critical software has implications for each of the areas NIST is discussing in this upcoming workshop. There are two we would like to specifically address:

1. First, system and component firmware should be explicitly called out in the guidelines outlining security measures that shall be applied to the federal government's use of critical software. This would clarify that traditional security measures, such as vulnerability management, patching, risk assessment, monitoring, and incident response, must also be applied to device and component firmware. Because of the privileged position occupied by firmware and its critical role in the software supply chain, it merits explicit mention similar to the SP 800-53 rev 5 updates.
2. Secondly, in the guidelines for software integrity chains, development, and provenance, it is particularly important to cover firmware update processes and include firmware as part of the Software Bill of Materials. Update processes have been targeted in the recent Sunburst attack and the ShadowHammer attacks. Visibility into component-level hardware and firmware, management of firmware vulnerabilities and updates, and validating the integrity of firmware are critical to defenses against these threats. The NIST Secure Software Development Framework can feature this protection in *PS.2.1: Make verification information available to software consumers* and *PW.3.2: Use appropriate means to verify that commercial, open source, and all other third-party software modules and services comply with the requirements*. By explicitly including firmware into these best practices, organizations will gain the visibility needed to dramatically change their risk posture.

The hidden ecosystem of firmware is no longer beyond the reach of enterprise tools that fit into common organizational best practices. Protection measures that are common in application software, such as signature checks, secure storage, execution integrity checks, and isolation, are exceedingly rare in firmware. There is, therefore, a compelling need for guidance for critical software to apply explicitly to firmware, the foundational layer of all software.