Secure Boot: An Effective, Low-Risk Alternative to Commercial Solutions

What You Will Learn

This paper describes:

- The various Secure Boot methods used in commercial computing and embedded system devices
- The applicability of these methods to network devices used in critical infrastructure
- The capabilities, technologies, risks, and benefits of these methods

We then describe a low-risk system level Secure Boot technique to reduce supply chain vulnerabilities while maintaining the same size, weight, and power parameters of the system.

Introduction

In today's world, where adversaries use the cyber domain and global manufacturing operations as platforms for attacks, protecting and maintaining the embedded systems that constitute the backbone of critical infrastructure is increasingly important. These systems carry critical communications and must have built-in robust defenses against cyber-attacks from the beginning of the product lifecycle. They must make sure that malicious adversaries have not tampered with the software and firmware of the systems' processing elements.

To satisfy this requirement, Secure Boot was deployed beginning in 2009 on Cisco® Smart Grid routers. As a system is powered on, Secure Boot prevents the code from being modified and protects the system against malware, logic bombs, and other nefarious instructions.

Secure Boot is established by anchoring trust in an element of the system that is rigorously controlled. The trust anchor establishes a foundation to begin the authentication chain and validate the integrity of the rest of the system. In this chain, each piece of software is authenticated by the previously loaded piece of software.

Threats, Risks, and Mitigations

IT systems are compromised daily by malware. The methods for introduction vary. For example, malicious websites can repeatedly infect a user's machine, and USB thumb drives can deliver malware into isolated systems. The common denominator of many threats is the ability of an adversary to persistently run code on a target system. The notion of persistence is important and reflects the capability of the unauthorized code (malware) to remain in the system even after the system is rebooted or reset.

In the past, malware has been observed in traditional IT systems with commercial operating systems and processors. More recently, Stuxnet and Duqu demonstrated the expanded capability of malware to create, deliver, and run unauthenticated, modified code on special-purpose systems that operate in isolated environments. [1]

Within most supply chains, the original equipment manufacturers (OEMs) rely on global manufacturing operations. The manufactured product is often transported across borders and goes through multiple storage depots. These transportation and logistical processes provide opportunities for nefarious entities to tamper with a product before it reaches the end user.

The most important way to mitigate all these threats and risks is to make sure the system boots only authenticated code. The root of the authenticity check is the single highly trusted anchor. This root helps recover compromised systems and helps prevent many injection scenarios.

Commercial Secure Boot Solutions

Secure Boot solutions exist in the commercial CPU marketplace today. A wide variety of techniques are used to build protections into or around the CPU.

One major Secure Boot technique focuses on "measuring" (that is, hashing) software as it boots. The actual measurement is done from a protected environment on the commercial CPU itself. The results are stored in a tamper-resistant storage device. This measurement process depends on an external system to subsequently query the measurements. The external system is then responsible for determining whether the hash is "good" or "bad" as well as for taking any action that results from this determination.

Another prevalent method focuses on an immutable anchor inside the CPU that has access to important material, also typically inside the CPU. The material is used to validate the authenticity and, in some cases, to decrypt the software to be run on the CPU. In this way, the system can deliver Secure Boot through CPU-based integrity checking of the software that is loaded into the CPU before it is run.

Challenges with Commercial Solutions

The two methods described above are used in many commercially available Secure Boot solutions, with a large number of variations in both the technical design and its implementation. As a result, each solution is CPU-specific. It provides different security characteristics and tradeoffs, and when implemented, it may generate inconsistencies across product lines. CPUs must therefore be chosen carefully to help ensure Secure Boot support.

In the newer Measured Boot, the system must interact with external systems for the measurements to be validated. The intricacies involved in implementing system recovery after failed measurements create significant challenges in embedded systems that use a Measured Boot approach.

The permanent anchors inside CPUs provide robust protection by introducing immutability into Secure Boot. However, a major exposure in most implementations of this method is a reliance on a single key, or at most two public keys, for validation of integrity or decryption of the software. Because the private keys used for signing or encrypting the software are on computer systems, the threat that a key will be compromised or inappropriately used is a real risk.[2] The CPU-based immutable anchors must therefore handle key revocation. Currently, no commercial solutions provide this important capability. Another major exposure in this method is the fact that the same CPU models are made with and without Secure Boot capabilities. Electronic component replacement attacks, where CPUs are swapped out, are real threats to the Secure Boot capability in the resultant system.

A System-Level Approach to Commercial Solutions

Because of the complexities associated with using commercial Secure Boot solutions, alternative solutions have been developed to address risks, maintain commerciality, and deliver a system that boots only genuine authenticated code. This system-level solution overloads an existing element such as a critical system field-programmable gate array (FPGA), application-specific integrated circuit (ASIC), or other component with Secure Boot logic. The logic maintains many protections in the Secure Boot system itself and validates the integrity and authenticity of software that the commercial CPU fetches to boot. If the system is deemed to have been trying to load invalid code, the critical system component has control over when to restart the system and, if needed, recover the system.

The system-level approach to commercial Secure Boot has the advantage of being CPU agnostic. The root of trust for integrity verification is outside the CPU itself. In a system-critical component, variability in CPU

capabilities can be safely ignored, so users can consider a greater selection of commercial off-the-shelf (COTS) processors. Since the system approach uses existing critical system components, control of the system is already in place. The likelihood of attacks that replace the electronic components to remove the Secure Boot capabilities is decreased. A system-level approach can be easily implemented in a way that supports key revocation with in-band or out-of-band management.

Smart Grid Routers

The Cisco 1240 Connected Grid Router supports reliable communications for field-area networks. Given the market requirements for this critical infrastructure device, we conducted research into the feasibility of adding a robust Secure Boot capability to the design. The platform and CPU could not be changed because of the limited availability of COTS parts that can perform well in harsh outdoor environments. The COTS CPU with the right environmental characteristics did not have native CPU-based Secure Boot capabilities, nor did it provide any mechanisms for Measured Boot.

These constraints led to a feasibility study about a system-level approach to Secure Boot. The study showed that, with no change in components and with an injection of the Secure Boot logic into a pre-existing programmable device, the design could be modified to support a Secure Boot implementation with little cost and engineering impact. The changes necessary for the addition of Secure Boot occurred within already qualified components and did not affect the readiness of the rugged COTS platform.

With this proven technology, Cisco has continued to implement Secure Boot across its product families, including routers, switches, appliances, and collaboration platforms as the criticality of digital infrastructures increase.

Conclusion

Secure Boot is a critical security capability. It helps thwart the introduction of untrusted software into a critical infrastructure device. It starts by building a root of trust at a basic level of the system and then validates the integrity and authenticity of each piece of software that runs on the platform. Secure Boot is especially beneficial in reducing the risk of global manufacturing operations and remote deployment.

The existing commercial techniques of Secure Boot are limited from the perspective of both security and availability. A systems approach is the most effective option for delivering a commercially viable, securely booting system.

For More Information

Visit the Cisco Trust and Transparency Center.

References

1. W32.Duqu: The Precursor to the Next Stuxnet.
   https://www.usenix.org/sites/default/files/conference/protected-files/leet12_duqu.pdf
2. Security Advisory: Revocation of Adobe code signing certificate.
   http://www.adobe.com/support/security/advisories/apsa12-01.html