ECU_102B_1_ Building Security into Your System: Protecting the Platform through Measurement and Attestation

Jeff Plank
Abstract

Securing the operational state of components has become an ever increasing topic among the industry. Much of the industry has secured the platforms upon which they operate but the sub components have become the next bastion of enforcing a security model. In this talk, we will cover the attack vectors and counter measures to head off the vulnerabilities in embedded firmware that previously appeared safe. We will discuss recent events, industry initiatives, the notion of trusted firmware and what users should look for in a secure device.

Learning Objectives:

1. Understanding the security landscape and what has ultimately changed in the industry
2. Threat modeling for the new age of protection
3. Understanding how secure trusted firmware translates into solution requirements and product guarantees
4. Learn what attestation measurements are and how they translate into proving to the platform what firmware is actually running
Agenda

- Threat model
- Secure boot for everyone
- Attestation, what happens after Secure Boot
- Reporting via DMTF PCMI Security Protocol
Trusted Platforms - Why the need?

- Various Points of entry
- Where has the product been?
- Is it really the expected product?
- Was it intercepted in flight?
- Is it running altered firmware / hardware?
- Does it contain the intended components?
- Will it stay that way?
- Is the product genuine?

Security Threats Along the Way of Manufacturing & Deploying

Flash Memory Summit 2019
Santa Clara, CA
What is Secure Boot?

- **Silicon HW Root of Trust**
  - Security begins with the Root of Trust contained in the ASIC
    - Embedded Signing Keys
    - Strong Hashing Functions
    - Immutable Authenticating Boot logic in Silicon Boot ROM

- **Board Components enablement and Security**
  - Trust is extended by verifying the authenticity and integrity of FLASH content prior to executing it
  - Digital signatures are supplied with all Firmware and Configuration Binaries
  - Validated with Embedded ASIC signing keys
  - ASIC Calculated Signatures are computed against the stored images and compared with stored signatures.
What is Attestation?

- On demand evidence that the **product** is configured and performing the function intended
- Usually report through a non-mutable mechanism but can be reported by trusted firmware
- A series of measurements taken during boot that reports HW / FW states of a device like a TPM using TCG Dice and Microsoft RIOT methodologies
- Can be used to detect old versions, new versions or rogue versions of FW
- Can also be used to detect the hardware state and authenticity of the part
- May be implemented as a reset of the HW for new measurements or a isolated security processor
- Platform Roots of Trust use attestation to continually monitor and validate system components
Attestation: Example Flow

- Public Attestation Keys
  - Response
  - Passes Measurement Valid State
  - Fails Measurement Take Countermeasures

- Attestation Requestor
- 1st Executable
- 2nd Executable
- Boot Loader

- Nonce Response
- Response Nonce
- 2nd Executable Measurement
- 1st Executable Measurement
- Security Values
- Strap Values
- Boot Loader Reset

- PKA Encryption Engine
- Private Attestation Key

Flash Memory Summit 2019
Santa Clara, CA

Flash Memory Summit
Manufacturing Identification and Authorization

- Authenticated Parts
- Protection from unauthorized adaptation
- ASIC Unique Identification
- ASIC Authorization / Registration to HSM
- Customer Customization with Public Key injection

Challenge/Response with Attestation Measurements
Who Measures: System of Trust

- BMC
- CPU
- ROT Security Processor
- SPI Flash
- Adapters
- Drive

Trusted Security Processor with Singular Function for Maintaining Trust
Measurement Reporting via PMCI MCTP

- DMTF is working on a protocol to first authenticate and then exchange a measurement from the components in a system in support of Attestation (RIOT).
- MCTP supports multiple attachment mechanisms (VDM and I2C by example).
- The protocol will allow for the endpoints to negotiate the supported algorithms, security protocol, and bit strengths.
- The exchange protocol has reached WIP release state and is ready for feedback.

https://www.dmtf.org/content/get-involved-dmfs-pmci-security-task-force

https://www.dmtf.org/content/pmci-security-architecture-wip-04-03-2019

https://www.dmtf.org/sites/default/files/standards/documents/DSP0274_0.9.0a.pdf

https://www.dmtf.org/sites/default/files/standards/documents/DSP0275_0.9.0a.pdf
Security of Platforms

- Grounded in signed secured firmware validated by unchangeable hardware
- Measured and reported by trusted firmware with unique measurements
- Aggregated in the platform by a discrete component providing coordinated measurement and actions to protect the platform from misuse or attack.
- Microchip provides both embedded security in its ASICs and platforms roots of trust to enable a secure platform.
Thank You
Security is Journey

Secure Software
Trust the OS

Secure Firmware
Trust the Firmware

Secure ASIC
Trust the ASIC

Security Continuum

MSFT Patch Tuesday (1998)

OpenSSL (2014)

Industry Enlightenment for Embedded NSA Intrusion (2015)

Intel Meltdown/Spectre SuperMicro (Barron) iLO4 HPE Home Routers (2018)

Next Threat Quantum Computing

Security for Server Components is Becoming Real
What is Secure FW Update?
Signed MSCC FW Validates Incoming FW Before Update

1. Adapter ships with signed MSCC FW or signed FW from the solution provider

2. Running Validated Firmware validates incoming images that have been signed by the vendor private key

3. Running Firmware updates the firmware in persistent storage

4. ASIC is reset again to validated new firmware.

5. New Firmware is up and running
What is Secure Debug Mode?

1. Hash of ASIC Attributes (Debug Token)
2. Signed (Hash of ASIC Attributes) (Debug Token)
3. Signature of Hash of ASIC Attributes

1. Unlocks Debug Ports
2. Accepts:
   - Signed Part specific images
   - Signed Production images which forces exiting of Secure Debug Mode.

- Request Debug
- Authorize Debug
- Validate Debug