TrenchBoot

Daniel P. Smith

Introduction

- Daniel P. Smith
  - Apertus Solutions Founder/Chief Technologist
- Plan
  - Review a few security concepts
  - Introduce TrenchBoot
  - Present current work underway
  - Open question and discussion

Security Concepts

Trust

- Definition (Merriam-Webster)
  - "**Assured** reliance on the character, ability, **strength**, or truth of someone or something"
- Simple mathematical relation
  - Trust $\propto$ Assurance of Strength $\propto$ 1 / Skepticism

Strength of Mechanism

- “Strength of mechanism refers to how well a specific approach may be expected to achieve its objectives. … Determination of strength of mechanism for each service has two components. The inadvertent threat and the malicious threat should be analyzed separately.”[sic][1]
- “The strength of mechanism is a relative measure of the effort (cost) required to defeat the mechanism and is not necessarily related to the cost of implementing such countermeasures.”[2]
- Essentially it is an assessment of how well it reduces the skepticism that the mechanism will fail through either direct or indirect application of force.

[1] TRUSTED NETWORK INTERPRETATION ENVIRONMENTS GUIDELINE, NCSCTG-11, National Computer Security Center

[2] Information Assurance Technical Framework, section 4.5.3, National Security Agency

Root of Trust

- A mechanism
  - Defined as “a process, technique, or system for achieving a result”
  - The action of achieving a result is the basis for the trust, i.e. the trust anchor
  - The degree of skepticism that the mechanism's result can be subverted imparts its Strength of Mechanism
- Selection should be based on the Strength of Mechanism required to meet assurance goal(s) of the system

Rooting the Trust

- Where is the mechanism implemented?
- Hardware based (higher strength)
  - is a mechanism that is implemented in hardware, e.g. CPU instruction
  - incorrectness can only be resolved by patching hardware (microcode) or manufacturing new hardware
- Software based (lower strength)
  - is a mechanism that is implemented by software, e.g. firmware
  - incorrectness can be resolved through flashing, software updater, etc

Roots of Trust

- Root of Trust for Storage (RTS)
  - a computing engine capable of maintaining an accurate summary of values of integrity digests and the sequence of digests[3]
  - provides a protected repository and a protected interface to store and manage keying material[4]
- Root of Trust for Measurement (RTM)
  - a computing engine capable of making inherently reliable integrity measurements[3]
  - provides measurement used by assertions protected via the RTI and attested to with the RTR[4]
- Root of Trust for Reporting (RTR)
  - a computing engine capable of reliably reporting information held by the RTS[3]
  - provides a protected environment and interface to manage identities and sign assertions[4]

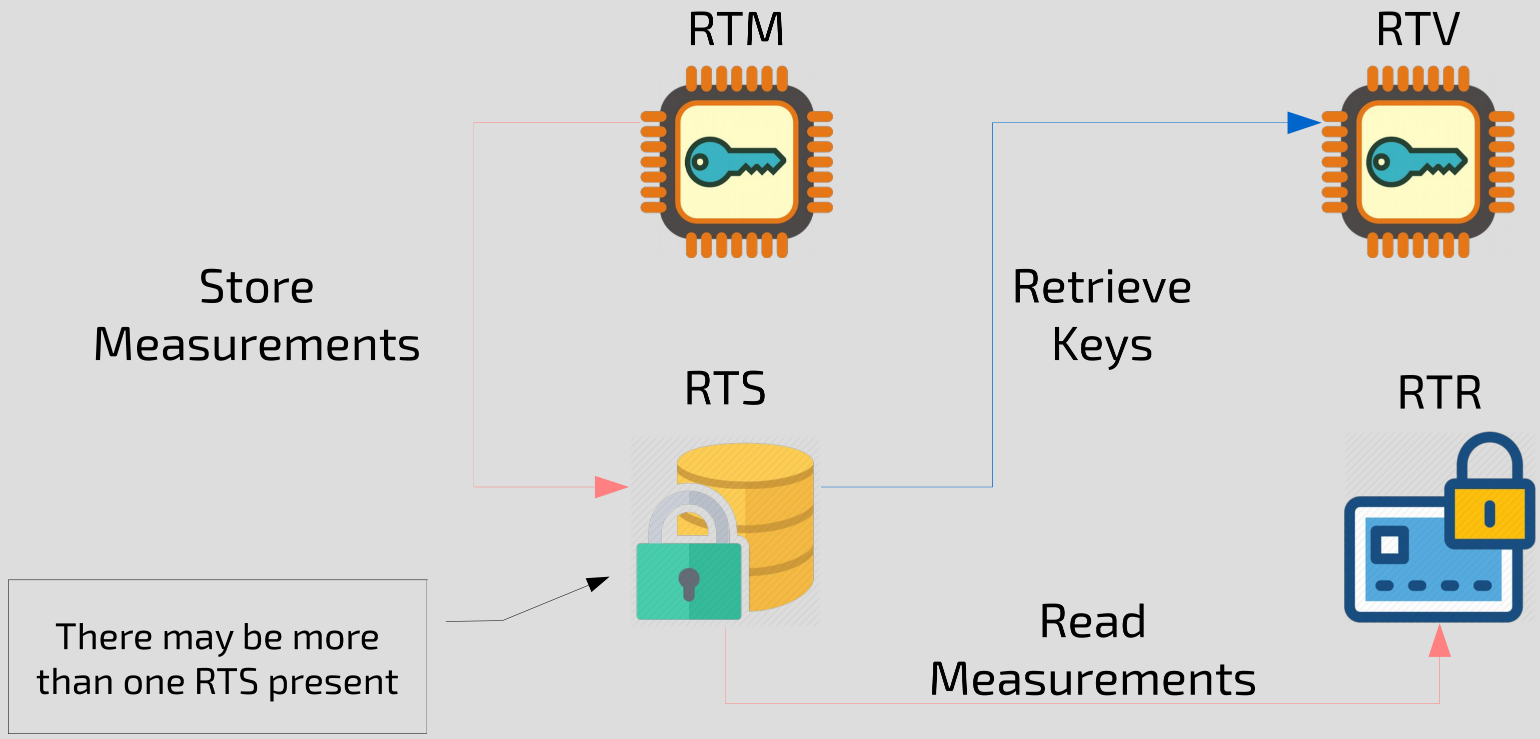
[3] TCG Specification - Architecture Overview, Section 4.2, Trusted Computing Group
[4] Guidelines on hardware-rooted security in mobile devices (Draft), NIST Special Publication 800-164

Roots of Trust

- Root of Trust for Verification (RTV)
  - provides a protected engine and interface to verify digital signatures associated with software/firmware and create assertions based on the results[4]
- Root of Trust for Integrity (RTI)
  - provides protected storage, integrity protection, and a protected interface to store and manage assertions[4]
- Other roots seen in literature
  - Root of Trust for Update (TCG)
  - Root of Trust for Confidentiality (TCG)
  - Root of Trust for Enforcement (UEFI)

[3] TCG Specification - Architecture Overview, Section 4.2, Trusted Computing Group
[4] Guidelines on hardware-rooted security in mobile devices (Draft), NIST Special Publication 800-164

Root of Trust Interactions



RTM

RTV

Store Measurements

Retrieve Keys

RTS

RTR

Read Measurements

There may be more than one RTS present

Measured Root of Trust System

- RTM Mechanism
  - Is an integrity scheme implemented with a cryptographic hash function
  - Strength is determined by what calculates the Core Root of Trust Measurement (CRTM)
- RTS and RTR Mechanism
  - Implemented with a TPM
  - RTS Strength is determined by TPM's protected capabilities and shielded locations
  - RTR Strength is determined by TPM's ability to establish identity and ensure authenticity of stored values in the shielded locations

Verified Root of Trust System

- RTV Mechanism
  - Is an authentication scheme implemented with asymmetric cryptographic signature
  - Implementation is software, often BootROM, loaded from fused flash (ARM) or SPI flash (ARM/x86)
  - Strength is determined by assurance that verification can not be subverted
- RTS Mechanism
  - Implementation varies, SPI flash (UEFI), e-fused on-chip flash (ARM), e-fused register/SPI flash combination (Intel BootGuard/ARM)
  - Strength is determined by write accessibility
- RTR Mechanism
  - Implementation with a TPM
  - Strength is determined by TPM's ability to establish identity and ensure authenticity of stored values in the shielded locations

Transitive Trust

- It is a mechanism similar to Root of Trust
  - Process to verify properties about an external (secondary) entity so that it may be included within the trust boundary
  - The result of extending the trust boundary is referred to as the trust chain or chain of trust.
  - Also has a Strength of Mechanism that has to be considered
- Verification comes in multiple forms (strengths)
  - Blind trust – transitive trust was establish therefore it is good
  - Measurement – evidence is collected and can be verified at any time

Attestation

- “Attestation is the activity of making a claim to an appraiser about the properties of a target by supplying evidence which supports that claim.”[5]
- Appraisal can be conducted on device or off device.
- Evidence can be collected at a variety of granularity
- Assertion should be bound to RTR
- Allows for verification to the validity of the trust chain

[5] Coker, George, et al. "Principles of remote attestation." International Journal of Information Security 10.2 (2011): 63-81.

Platform Launch

- A launch is when a platform starts from quiescent state and progresses to executing an end run-time, e.g. a hypervisor or an operating system. There are two types of launches,
  - Static Launch: when the platform CPU comes out of reset, in other words the CPU was either powered on or hard/soft power cycled
  - Dynamic Launch: when the platform CPU is re-initialized as if it came out of reset without actually putting the CPU in reset

Boot Integrity Technologies (BITs)

(1) Hardware Assisted Boot

(2) Intel Boot Guard

(3) TCG Bios

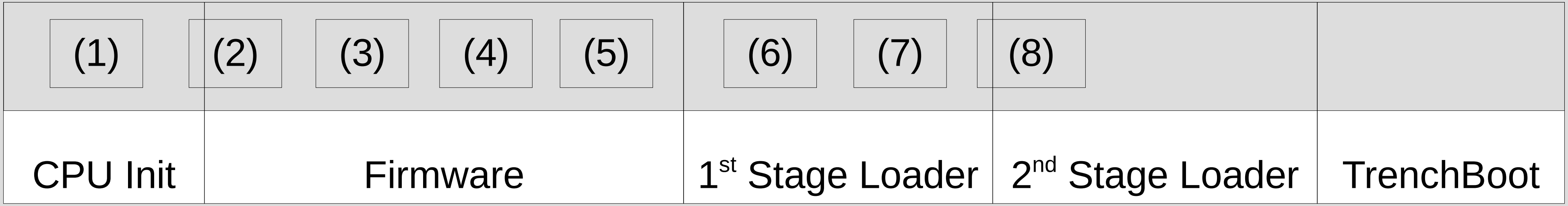
(4) UEFI Secure Boot

(5) Open Systems Firmware

(6) Trusted Grub

(7) RedHat UEFI shim

(8) DRTM (Intel TXT/AMD SL)



(1) (2) (3) (4) (5) (6) (7) (8)

CPU Init

Firmware

1st Stage Loader

2nd Stage Loader

TrenchBoot

TrenchBoot

Objective

A project focused on delivering a uniform, cross-platform framework to enable open platform builders the ability to securely incorporate BITs.

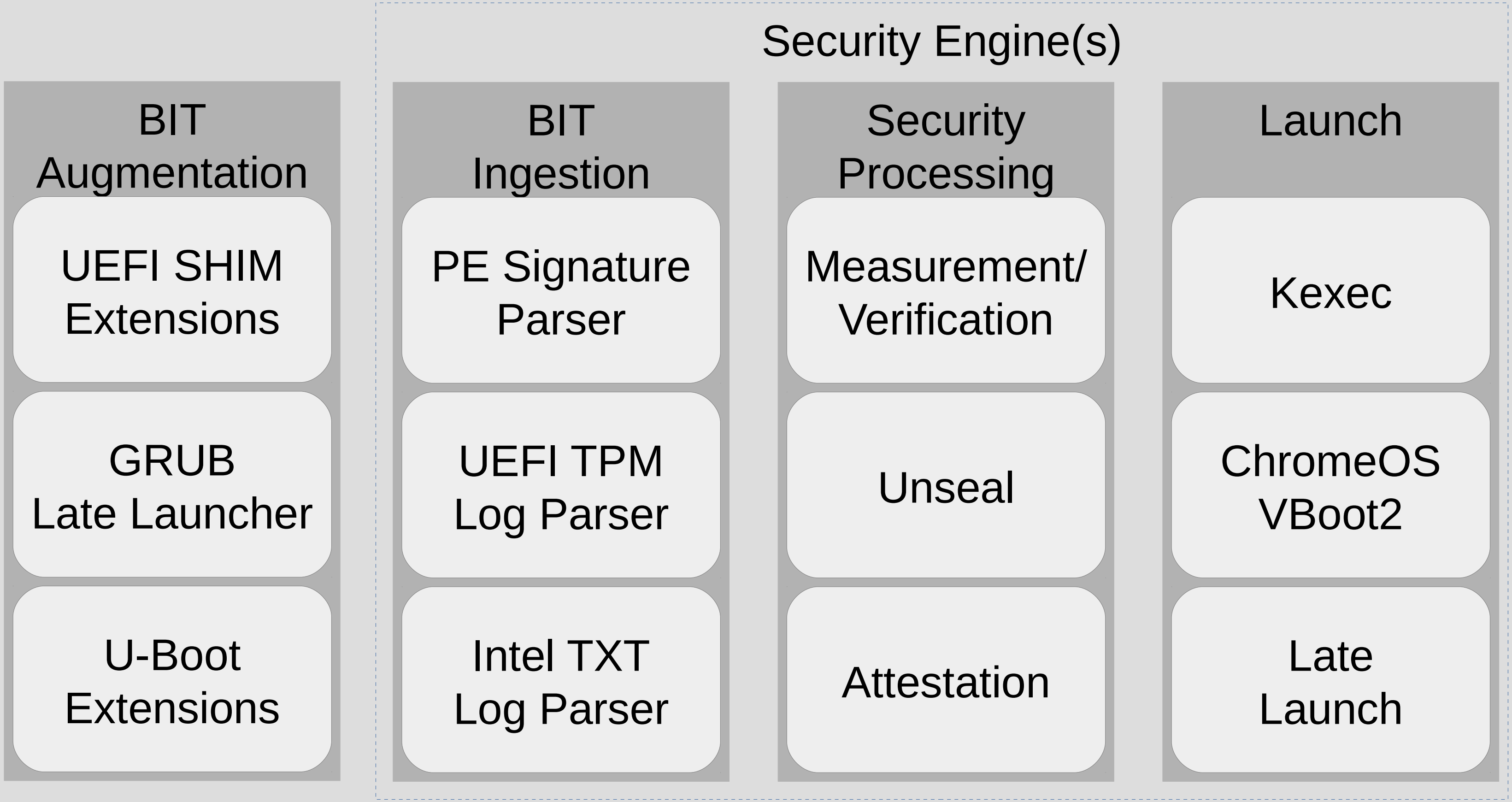
Inspiration

- In 2015 started work on improving the “Measured Launch” capability in OpenXT
  - Found myself constrained retrofitting into a highly coupled solution (there was success in decoupling)
  - The enforced integrity was limited/brittle, platform had to be in one exact state otherwise it was considered unknown
  - The solution was constrained to Intel TXT (VPro) platforms, typically only found in an OEM's enterprise class product lines

Long Term Goals

- A common framework that builds purpose driven security engine
  - Provide an abstraction of supported BITs' results
  - Provide a common set of security actions, a subset processing BITs' results
  - Provide the ability to configure a sequence of security actions
  - Leverage OpenEmbedded build system to provide configurable, reproducible builds
  - Support a variety of use-cases
- A source of ground truth
  - Documented Strength of Mechanism evaluation/explanation for supported BITs
  - Collection of crowd-sourced and validated database of measurements
  - Provide a publicly available attestation service

General Architecture



	Security Engine(s)		
BIT Augmentation	BIT Ingestion	Security Processing	Launch
UEFI SHIM Extensions	PE Signature Parser	Measurement/ Verification	Kexec
GRUB Late Launcher	UEFI TPM Log Parser	Unseal	ChromeOS VBoot2
U-Boot Extensions	Intel TXT Log Parser	Attestation	Late Launch

Current Effort

Embedded AMD OpenXT Server

- A demonstration of using AMD late-launch to implement a new measured secure boot to launch OpenXT
- Consist of,
  - PCEngines APU2 retrofitted with TPM 1.2
  - GRUB late-launch relocator
  - LinuxBoot SecureLoader/u-root Security Engine
  - Kexec-able OpenXT Xen

A New Approach to Secure Boot

- Traditionally
  - Verification → Measurement → Attestation
  - Ownership passed by blind delegation
  - Delegated Key tampering difficult to detect
- Measured Secure Boot with Grub Late Launch (AMD)
  - Measurement → Verification → Attestation
  - Ownership passed through reprovisioning
  - Delegated Key measured/stored by hardware

AMD SKINIT Instruction

- SKINIT is available on every processor that supports AMD-V
- Unlike Intel, AMD left "Secure Loader" (SL) as an exercise for the implementer
- The SL Header is implementation specific
- No requirement that the SKINIT Hash Area (Length), includes all of the SL Code and Static Data

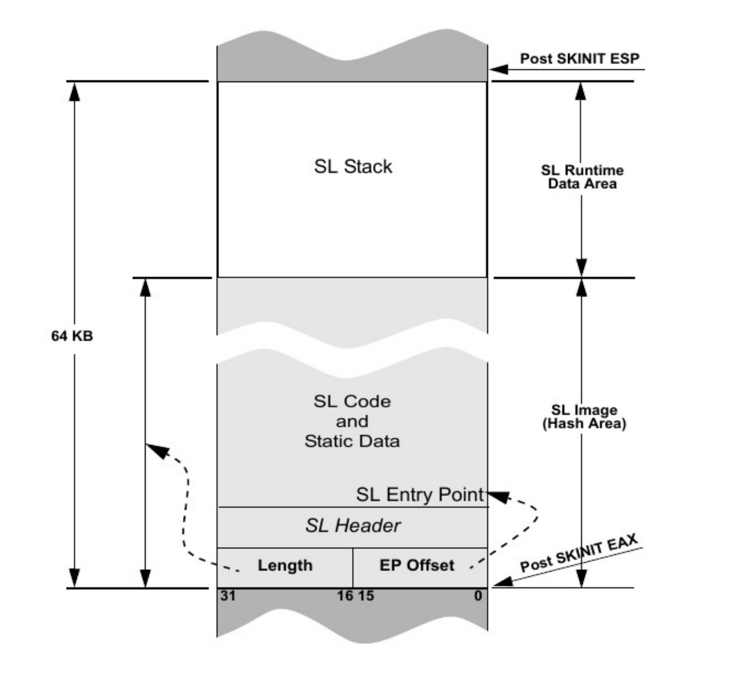
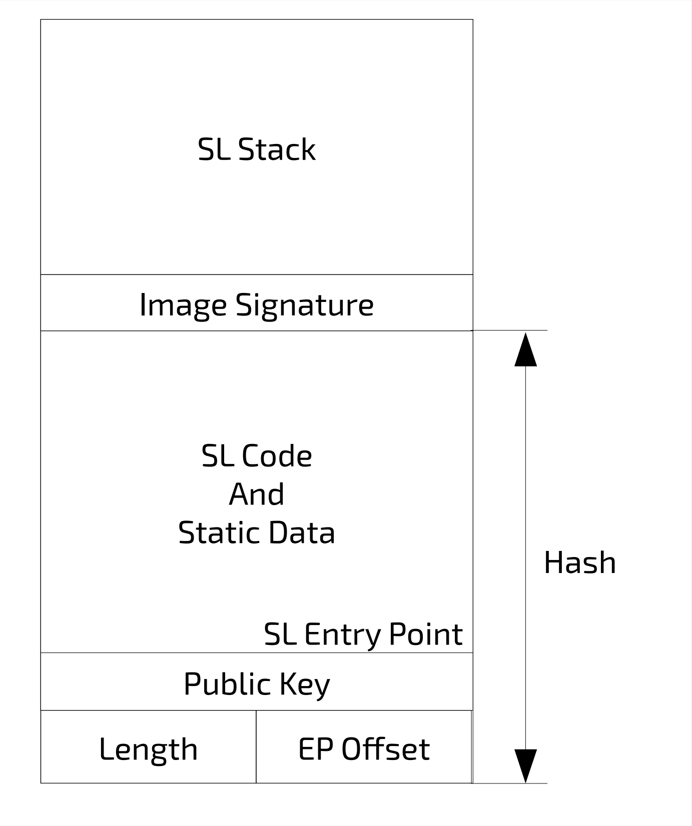


Fig. - AMD64 Architecture Programmer's Manual Volume 2: System Programming

Measure the RTV

- RTM and RTR are rooted in hardware with SKINIT and TPM
- RTV and RTS for Validation (key protection) is contained in CRTM



Benefits of the Approach

- Near impossible for CRTM to be subverted
  - Compare to Verification → Measure, where flash resident RTV has demonstratively been subverted
- Strong ownership without blind trust
  - Device specific binding through TPM provisioning
  - Target run-time can be sealed to expected measurements
  - Transfer is a matter of accepting new measurement, i.e. does not rely on a blind trust
- Strong and flexible key management
  - Supports enterprise key or per device key
  - Unlimited revocation
- Generally applicable approach

Questions