# Secure Boot, Measured Boot, and TrenchBoot

## An Introduction for the Xen Community

Daniel P. Smith
Apertus Solutions, LLC

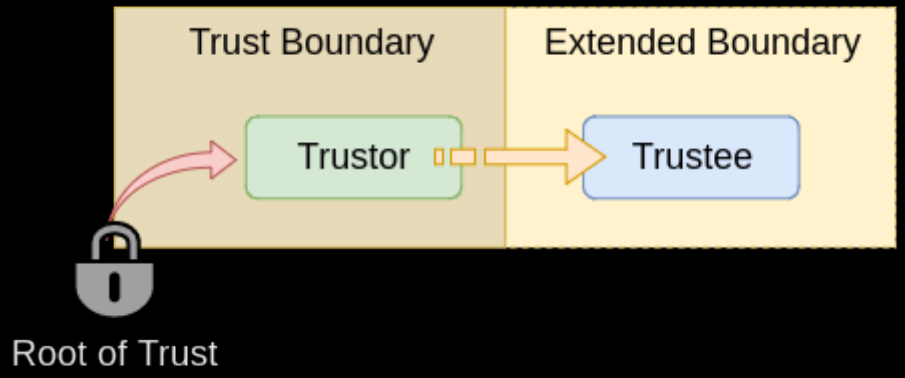
September 12th, 2018

# Vocabulary

- Strength of Mechanism
  - Ability of a mechanism to resist intentional or unintentional failure
- Root of Trust
  - An idempotent mechanism whose result asserts a fact or property
- Static Launch
  - A system launch that is a one time execution with the execution code at a fixed location in memory
- Dynamic Launch
  - A system launch that can be done repeatedly with the execution code able to reside at different locations in memory

# Transitive Trust

An operation conducted by a trustor that consists of one or more mechanisms used to assess one or more facts about a trustee before allowing the trustee to be included within the trustor's trust boundary and delegated the authority to act as a trustor.



# Secure Boot

- Secure Boot is a bit nebulous, will focus on what is implemented in UEFI
- UEFI consists of two launch Roots of Trust to build transitive trust chains
  - Verification Root of Trust – signature verification
  - Measurement Root of Trust – measurement collection
- The Verification RoT is what most refer to when speaking of Secure Boot
  - Relies on Boot Flash as Root of Trust for Storage to protect key database
  - Verification is not checked until DXE not during SEC, to quote Microsoft
    - "The Security terminology is actually a misnomer as there is no security verification of codes executing in this phase. It is assumed that malware cannot target the Flash as the Flash update can happen only through signature verification."*

* https://answers.microsoft.com/en-us/windows/forum/windows8_1-security/uefi-secure-boot-in-windows-81/65d74e19-9572-4a91-85aa-57fa783f0759

# Measured Boot

- A system launch where each component involved is measured through a transitive trust sequence, i.e. trust chain
  - Root of Trust for Measurement (RTM): The first component in the transitive trust chain
  - Core Root of Trust Measurement (CRTM): This is first measurement which is generated by the RTM
- The purpose is to record what firmware/software was used to launch the platform

# Types of Measured Boot

- Two common implementations of x86 Measured Boot are differentiated by the Root of Trust used to establish the trust chain
  - Static Root of Trust (typically rooted in mutable software*)
  - Dynamic Root of Trust (rooted in hardware)
- The two Roots of Trust are driven by the type of launch used
  - Static Launch → Static Root of Trust
  - Dynamic Launch → Dynamic Root of Trust

* Intel BootGuard and AMD GuardMI provide hardware firmware verification

# Static Launch Measured Boot

- When the RTM, referred to as the Static Root of Trust for Measurement (SRTM), is implemented as part of the static launch
  - For UEFI this is implemented in the Pre-EFI Init (PEI) software loaded from SPI system flash.
  - The PEI code hashes itself as the CRTM for the trust chain and then hashes DXE
  - Turned out corrupting boot flash, and thus the RoT, was much easier than expected. Intel addressed with Boot Guard and AMD provided Hardware Validated Boot under GuardMI to move the RoT into the hardware.
  - Recent developments (Boot SPI Interposer),
    - Trusted Computing Group DICE specification with solutions (MS Cerberus)
    - Google Titan chip
    - HP Sure Start

# Dynamic Launch Measured Boot

- When the RTM, referred to as the Dynamic Root of Trust for Measurement (DRTM), is implemented as part of a dynamic launch immediately following the static launch.
  - Achieved using the Intel GETSEC[SENTER] or AMD SKINIT instruction
    - Details can be found in TrenchBoot's documentation project*
  - The CPU hashes the dynamic launch software as the CRTM for the trust chain
  - Important motivations
    - Provides a method to address the "measurement gap" in the SRTM chain
    - Shortens trust chain to reduce fragility in the chain



* https://github.com/TrenchBoot/trenchboot/blob/master/documentation/Late_Launch_Overview.md

# Dynamic Launch Repeatability

- The x86 Dynamic Launch capability can be reused at runtime to reset the DRTM and build a new trust chain.
- Provides for multiple use-cases
  - Launch secure app in compromised environment (Flicker*)
  - Kexec-like measured launch

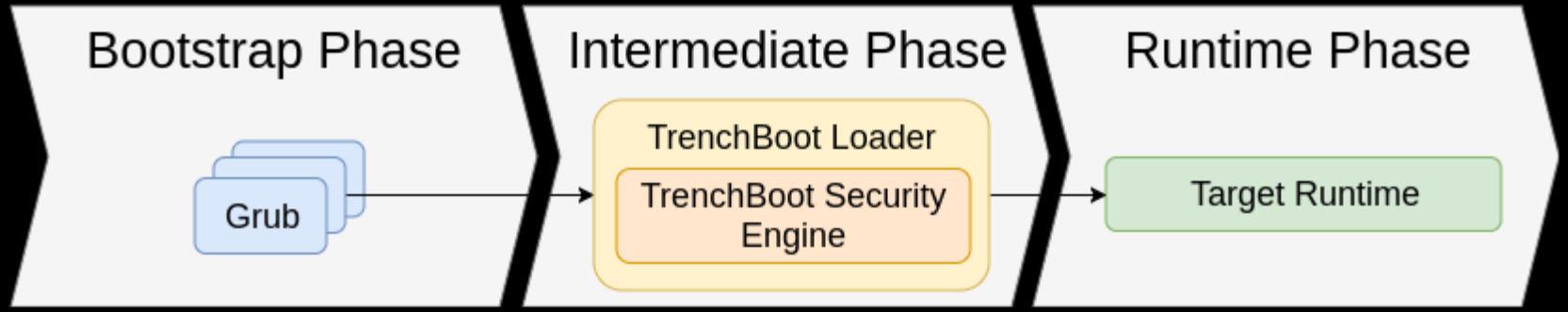
** McCune, Jonathan M., et al. "Flicker: An execution infrastructure for TCB minimization." ACM SIGOPS Operating Systems Review. Vol. 42. No. 4. ACM, 2008.*

# Why Care About Measurements

- They provide an accurate depiction of what software was and is in control over system, i.e. the system's state history
  - Verification solutions can only assure provenance
  - Collection through transitive trust transactions provides a chaining of measurements
  - This is what drives MS Zero Trust networks with 365 by enabling a device to communicate its state
  - Powers the “tamper-evident attestation claims” provided by Google's Shielded VMs.

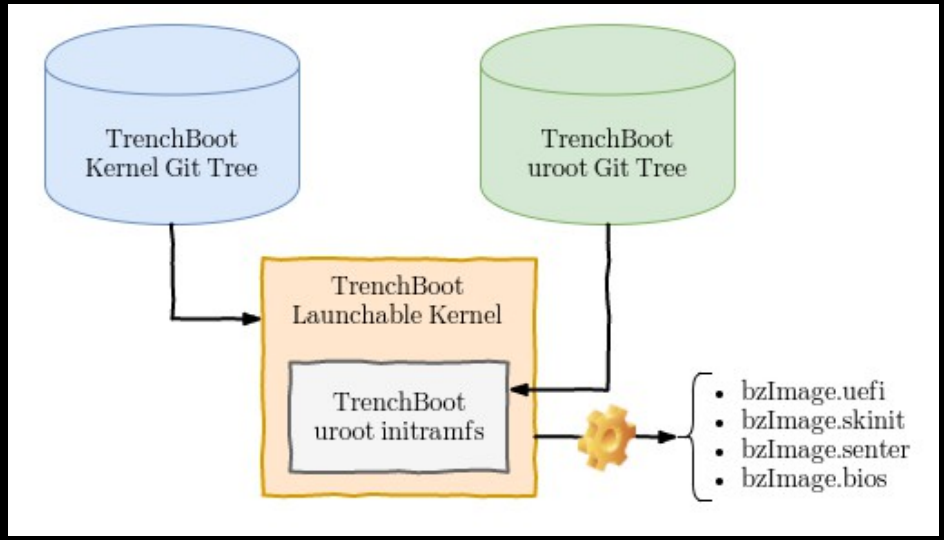
# TrenchBoot Overview

- TrenchBoot is a cross-community integration project focused on enabling constructing security engines to perform launch integrity actions for their systems.
- It functions by introducing an Intermediate Phase, similar to UEFI shim, tboot, etc., to the system launch life-cycle
  - Within the Intermediate Phase is the Intermediate Loader, TrenchBoot Loader, which is capable of being launch by a variety of bootstrap solutions.
  - TrenchBoot Loader contains the Trenchboot Security Engine that implements integrity processing



# TrenchBoot Loader Components

- Composed of well known components, Linux and u-root
- A TrenchBoot Loader consists of a TrenchBoot enhanced Linux kernel with a built-in TrenchBoot u-root initramfs
- TrenchBoot Security Engine is implemented as extensions to u-root
- Result is an image that can be launched by different boot mechanisms/environments



# Benefits of TrenchBoot

- Linux and u-root provide single source, cross architecture support, e.g. x86, ARM, s390x, PPC, MIPS
- Unifies system launch from a secure start-up point of view
  - Possible and future target includes VM launch
- Provides a rich environment for launch integrity inspection and verification process,
  - Off platform attestation, TCG Attestation SNMP MIB
  - Off platform key retrieval, e.g. KMIP, YubiKey
  - Fine grained attestation

# Xen Support Sought

- TrenchBoot uses kexec to launch target runtime
  - Xen kexec entry needs to be able to be invoked from an environment initialized by either BIOS or UEFI post ExitBootServices
  - Consideration is removal of dependency on UEFI Boot and Runtime Services
- TrenchBoot will need to pass information to Xen for launch
  - Launch vector: BIOS/CSM, UEFI, TXT, AMD-V
  - Memory reservation table: e820 or UEFI memory map
  - Integrity information: measurement log, key material, etc
- TrenchBoot aware runtimes will need to retrieve launch information
  - Will need to collaborate on Guest ↔ Hypervisor communication (hypercall?)

# Use Cases for Xen

- Provides a cross platform unified secure launch vector
- Enables Xen to be compatible with LinuxBoot variations of Open Compute Open System Firmware
  - TrenchBoot chose u-root to align with firmware project LinuxBoot
  - TrenchBoot extensions are targeting upstream u-root and LinuxBoot

* https://www.platformsecuritysummit.com/2018/speaker/hudson/