

Paving the Path for TrenchBoot DRTM in Xen Hypervisor





Xen Project Developer and Design Summit 2023

Michał Żygowski





Michał Żygowski
Firmware Engineer

-  [@_miczyg_](https://twitter.com/_miczyg_)
-  michal.zygowski@3mdeb.com
-  linkedin.com/in/miczyg
-  facebook.com/miczyg1395
- Braswell SoC, PC Engines and Protectli maintainer in coreboot
- OpenPOWER System Software Technical Workgroup chair
- dedicated to the open-source firmware since 2017
- interested in advanced hardware and firmware security features

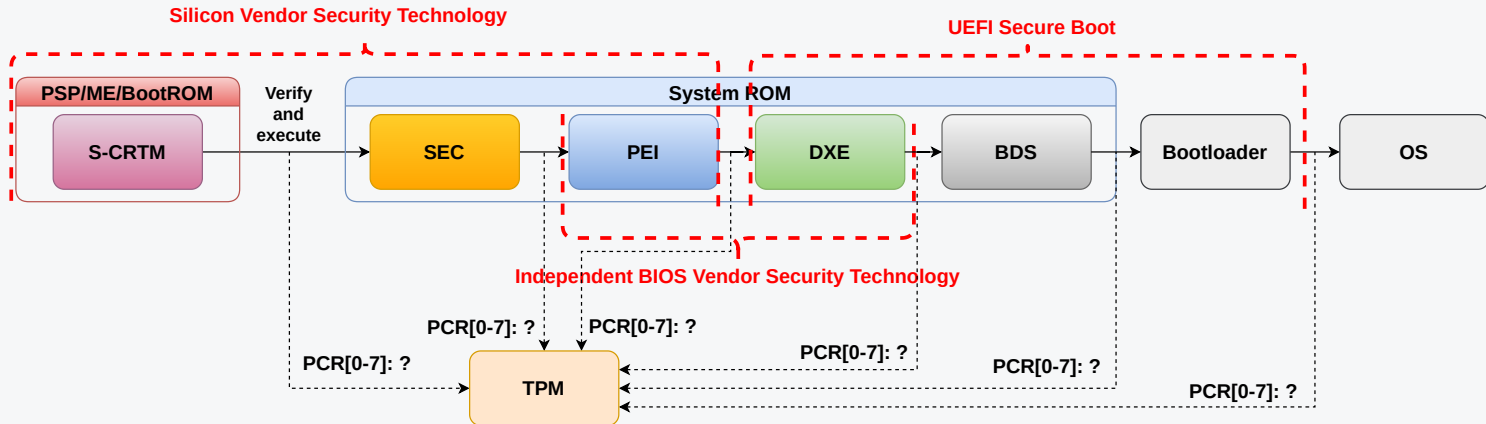


- coreboot licensed service providers since 2016 and leadership participants
- UEFI Adopters since 2018
- Yocto Participants and Embedded Linux experts since 2019
- Official consultants for Linux Foundation fwupd/LVFS project since 2020
- IBM OpenPOWER Foundation members since 2020

- D-RTM vs S-RTM
- What is TrenchBoot?
- TrenchBoot vs TrustedBoot
- Usecase: Qubes OS Anti Evil Maid
- Q&A

- S-RTM
 - UEFI Secure Boot + Measured Boot
 - Silicon Root of Trust:
 - Intel Boot Guard
 - AMD Platform Secure Boot
- D-RTM
 - Intel Trusted Execution Technology (TXT)
 - AMD Secure Startup (SKINIT instruction)

Why bother with D-RTM if we have S-RTM out of the box in the BIOS/firmware?



- How hard it is to maintain firmware updates for correctly deployed S-RTM protection?
 - modern x86 platform may have 20 different keys and certificates
- How useful are information gathered during measured boot?
 - event log quality
 - forward sealing and updates



- It depends on mechanism delivered by platform and/or silicon vendor
- Verified Boot-like technologies (Intel Boot Guard, AMD Hardware Validated Boot, NXP High Assurance Boot etc.)
 - are those technologies open?
- How hard it is to sign firmware and fuse/provision platform?
 - it really depends on vendor

	Documentation	Application note or guides	Fusing tool	Signing tool	Feasible to use without NDA?
Intel	NDA required	NDA required	NDA required	NDA required	no
AMD	NDA required	NDA required	N/A	NDA required	no

Unfused platforms may disable Boot Guard

Vendor Name	ME Access	EC Access	CPU Debugging (DCI)	Boot Guard	Forced Boot Guard ACM	Boot Guard FPF	BIOS Guard
ASUS VivoMini	Disabled	Disabled	Enabled	Disabled	Disabled	Disabled	Disabled
MSI Cubi2	Disabled	Disabled	Enabled	Disabled	Disabled	Disabled	Disabled
Gigabyte Brix	Read/Write Enabled	Read/Write Enabled	Enabled	Measured Verified	Enabled (FPF not set)	Not Set	Disabled
Dell	Disabled	Disabled	Enabled	Measured Verified	Enabled	Enabled	Enabled
Lenovo ThinkCentre	Disabled	Disabled	Enabled	Disabled	Disabled	Disabled	Disabled
HP Elitedesk	Disabled	Disabled	Enabled	Disabled	Disabled	Disabled	Disabled
Intel NUC	Disabled	Disabled	Enabled	Disabled	Disabled	Disabled	Disabled
Apple	Read Enabled	Disabled	Disabled	Not Supported	Not Supported	Not Supported	Not Supported

Alex Matrosov 2017: BETRAYING THE BIOS: WHERE THE GUARDIANS OF THE BIOS ARE FAILING

MSI Boot Guard keys leaked for multiple devices

**Alex Matrosov** ✓

@matrosov



🔗 Recently, @msiUSA announced a significant data breach. The data has now been made public, revealing a vast number of private keys that could affect numerous devices.

🔥 FW Image Signing Keys: 57 products

🔥 Intel BootGuard BPM/KM Keys: 166 products

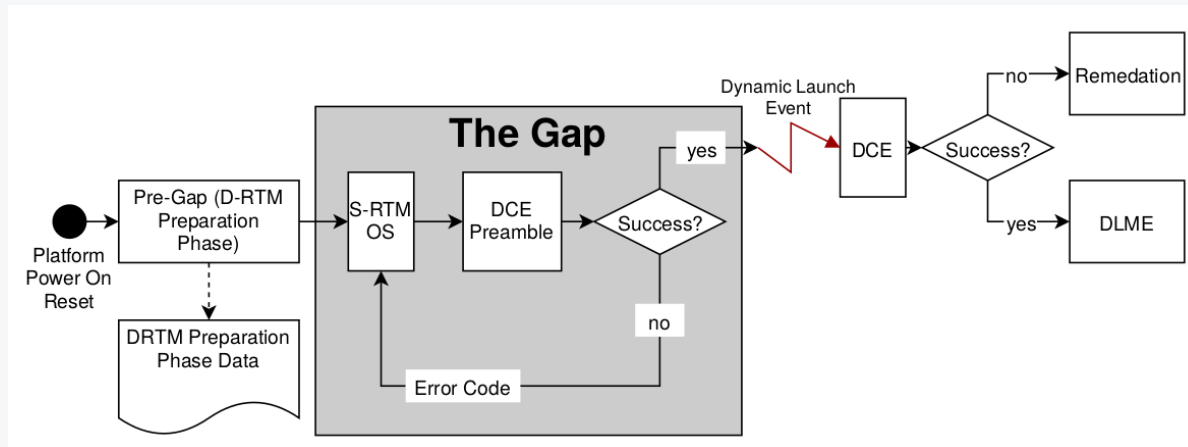
 github.com/binary-io/Sup...

Source: <https://twitter.com/matrosov/status/1653923749723512832?s=20>

- BootHole
 - Classic buffer overflow
 - What's next? BootHole v2? v6...?
 - Having a centralized Root of Trust in Microsoft's CAs is a problem
 - If Microsoft signs a buggy application, then it's over, millions of devices affected
 - Having a Microsoft signed Shim already allows booting potentially buggy self-signed application

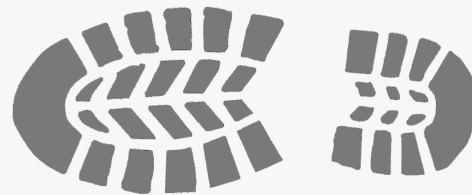


- **Dynamic Root of Trust for Measurement (D-RTM)** - can reestablish the trust at (theoretically) any moment of platform uptime
- The only prerequisite is the BIOS to initialize the D-RTM technology properly (only applicable to Intel TXT, AMD does not need special initialization from BIOS side)
- Performs secure measurement of the environment responsible for launching the target software/operating system
 - Lesser complexity -> less things can go wrong
 - Less components to measure -> simpler event log, easier forward sealing and updates management
- Trust rooted only in the silicon/microcode (at least for x86 architecture)



- **D-RTM Configuration Environment (DCE) Preamble** - responsible for platform configuration and initiating **Dynamic Launch Event (DL Event)**
- **D-RTM Configuration Environment (DCE)** - entered via **DL Event**, performs DLME measurement and hands off the control to it
- **Dynamically Launched Measured Environment (DLME)** - the ultimate result of the dynamic launch, which effectively can be OS or bare metal software
- D-RTM does not care about initial measurements since it use PCR[17-22] which are locked until DL Event unlocks them in TPM locality 4

TrenchBoot



- [TrenchBoot Mailing List](#)
- The #trenchboot channel on [OSFW Slack](#)
 - Also bridged to Matrix as [#OSFW-Trenchboot:matrix.org](#)
- Twitter [@TrenchBoot](#)

TrustedBoot

- TrustedBoot (tboot) supports only Intel TXT
- Is an exokernel and Xen (or any other kernel) has to be aware of its presence

TrenchBoot

- Aims for unified approach supporting both AMD and Intel processors
- The goal is to implement a native support for D-RTM to let Xen have full control without any exokernels



- Qubes OS Anti Evil Maid (AEM) is a set of software packages and utilities to aid against [Evil Maid attacks](#)
- Leverages DRTM and TPM to seal secrets, which are used by the owner to confirm whether the device has been tampered with or not
- Currently only Intel TXT and TPM1.2 with tboot is supported using legacy BIOS boot mode

Intel support

- Proof of Concept replacing tboot with TrenchBoot in the current AEM
- Implemented support for booting with Intel TXT and TPM 1.2 in legacy BIOS boot mode
 - GRUB multiboot module extended with preparing the TXT environment for secure launch
 - Xen extended with proper AP cores bring up and handling the boot flow after the secure launch
 - Xen extended with basic TPM code to measure dom0 kernel and initrd and record the measurements in the event log

AMD support

- [Patches](#) handling the CPU bring up on AMD were merged some time ago
 - Proof of Concept of TrenchBoot DRTM on AMD has been shown on [Qubes OS Summit](#)

- Preparing the PoC patches for upstream
- Extending the Qubes OS AEM with TPM 2.0 support
- Integrating the AMD DRTM support in Qubes OS packages

Project sponsored by [NLnet](#)

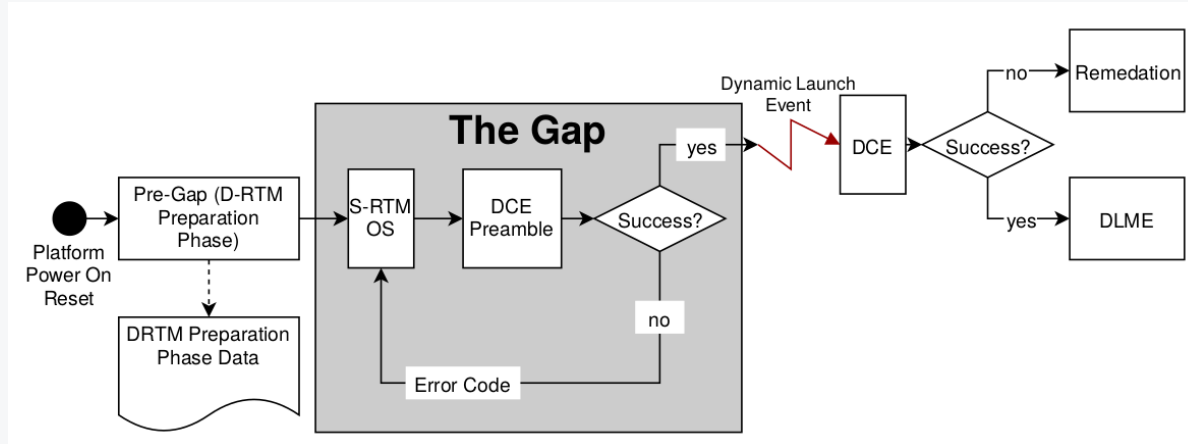


Qubes OS patches:

- <https://github.com/QubesOS/qubes-vmm-xen/pull/160>
- <https://github.com/QubesOS/qubes-grub2/pull/13>

Project description

- <https://docs.dasharo.com/projects/trenchboot-aem-v2/>
- <https://github.com/TrenchBoot/trenchboot-issues/milestones>



- **DCE Preamble** - reference implementation in GRUB
 - [Intel TXT RFC patches on grub-level ML](#)
 - GRUB performs platform specific configuration for DCE to perform DL Event
 - Intel TXT: **Measured Launch Environment Developer's Guide**
 - AMD Secure Startup: **AMD64 Architecture Programmer's Manual** (Vol. 2, chapter "Secure Startup with SKINIT")

DCE is the first component measured after DL Event

DL Event opens TPM's locality 4 and the CPU microcode (GETSEC[SENDER] or SKINIT instruction) sends the DCE to the TPM to be measured by the means of HASH_START/HASH_DATA/HASH_END registers.

- Intel TXT:
 - SINIT Authenticated Code Module (ACM)
 - Modules per microarchitecture can be publicly obtained from [Intel's portal](#)
- AMD Secure Startup:
 - Secure Loader was left to be implemented by developers
 - Reference implementation: [TrenchBoot Secure Kernel Loader \(SKL\)](#) (do not confuse with Skylake), formerly [Landing Zone](#)

Xen is the target DLME of the D-RTM. D-RTM implications on Xen:

- Intel TXT requires a Measured Launch Environment header to be present in binary (added to `head.S` source)
- To respect the measure-before-use rule, TPM code has been added to Xen to measure the multiboot2 MBI structure and dom0 components
- Intel TXT requires special CPU wakeup routine (DCE left the CPUs in different state than usual) which has been implemented in the early assembly code executed after entry to Xen
 - It is required for the standard CPU startup routines to work properly in Xen
- Xen needs to search for the TPM event log prepared by the DCE preamble to fill in the measurement events done by Xen explicitly
- Reserving platform specific memory ranges (Intel TXT register space, etc.)

Some of the above were being done by either tboot itself or by the tboot support code in Xen.

- Daniel P. Smith
- Andrew Cooper
- Daniel Kiper
- Krystian Hebel
- Ross Philipson

Q&A

Backup

UEFI approach (work in progress)

- For UEFI boot mode the DCE Preamble will set up a generic code module pointed to by EFI System Table Configuration Tables
- The module has a well [documented ABI](#)
- The target software will be able to get the module from EFI System Table and perform EFI Exit Boot Services (it was a hard requirement from Linux)
- Target software will fill the information for DCE and call into the module, which will perform DL Event

Future planned work

- Xen will be able to initiate the DL Event via the code module exposed by DCE Preamble (GRUB)
 - Although Xen does not use that much information from UEFI we want to unify the flow with TrenchBoot implementation for Linux kernel
 - Native EFI PE entry point shall be supported for UEFI Secure Boot purposes
- Xen will need to obtain the module from Secure Launch Resource Table linked in the EFI System Table Configuration Tables
- Xen will fill the necessary information for the DCE (like Xen address in memory to be measured) and call the module

More details on trenchboot.org