S-RTM and Secure Boot for VMs

Qubes OS mini-summit 2021

Piotr Król

3MDEB

whoami

Piotr Król
3mdeb Founder & CEO

- coreboot contributor and maintainer
- Conference speaker and organizer
- Trainer for military, government and industrial organizations
- Former Intel BIOS SW Engineer
- 12yrs in business
- Qubes OS user since 2016
- C-level positions in

3MDEB LPN PLANT VITRO

Who we are ?



- 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
    - Our Firmware Engineer Michał is chair of SSWG since 2021

Agenda

- Presentation goal and terminology
- S-CRTM
- What happen over last year in S-RTM and Secure Boot
    - key takeaways
- Challenges
- Q&A

Presentation goal

To discuss state of integration of various S-RTM and verified boot technologies for virtual machines

- There are many topics involved and we need some checkpoints to gather and discuss how things going on
- It is always good to have resources where developers can quickly grasp what is current state of the art

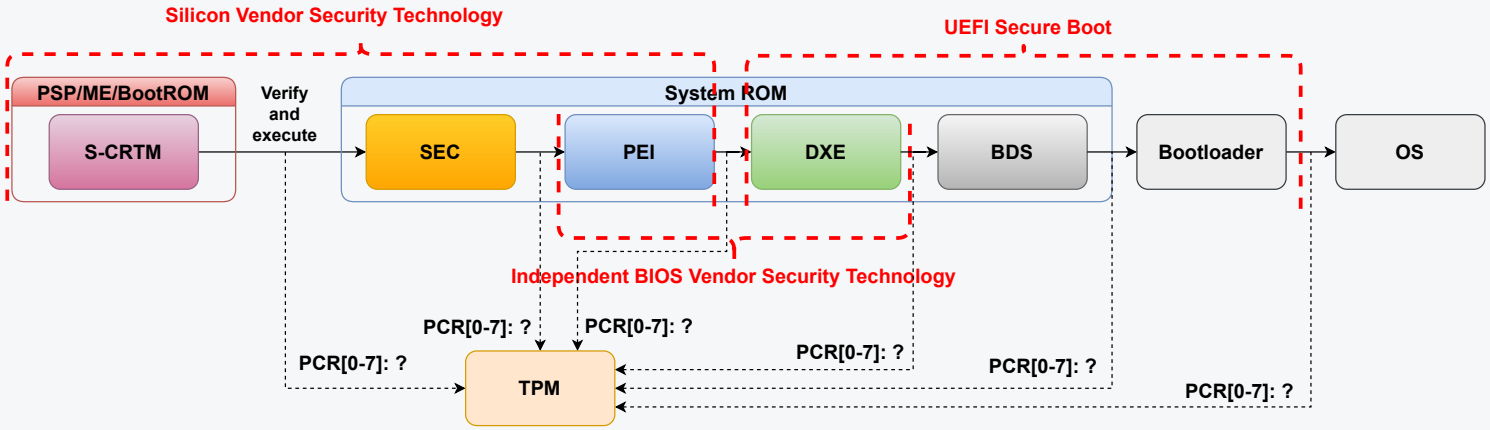
Little bit about terminology

- **S-RTM** (*Static-Root of Trust for Measurement*) in reality would be either **S-CRTM** (*Static-Code/Core Root of Trust for Measurement*) or **S-HRTM** (*Static-Hardware Root of Trust for Measurement*)
    - static in this case means point in time in contradiction to dynamic, which would be arbitrary point in time of platform operation
    - we are not aware of any widely available S-HRTM (despite many pretend to be implemented in hardware)
- **Secure Boot** was used just for presentation marketing and general understanding of topic
    - **Secure Boot** term is technically imprecise what will be explained later
    - we will use verified boot term
- Depending on situation if Root of Trust is used for Measurement or/and Verification, we should use S-CRTM, S-CRTV, S-CRTMV

Little bit about terminology

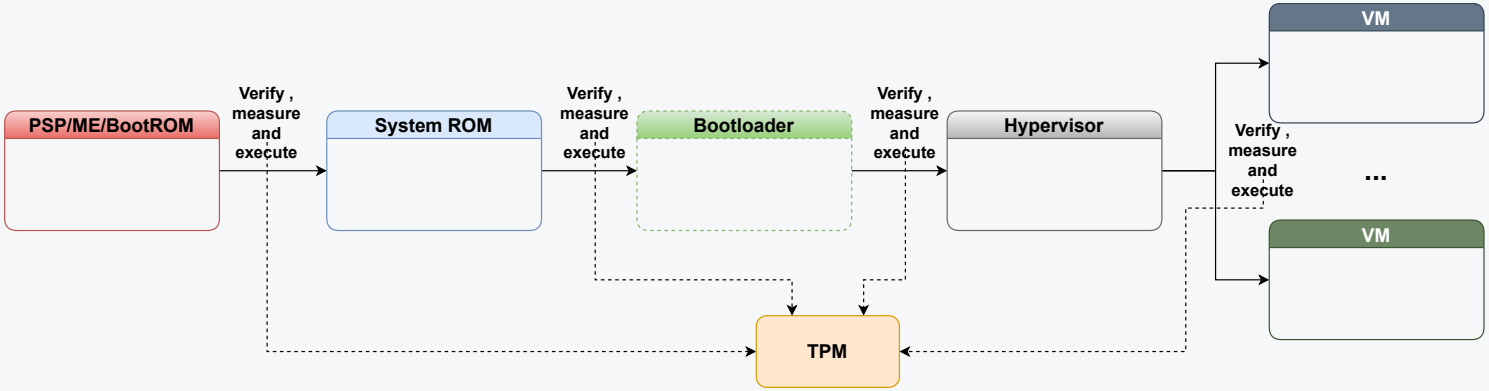
- System ROM - place where boot firmware is stored
- Boot process according to Platform Initialization (PI) and Unified Extensible Firmware Interface (UEFI) specifications, which are controlled by UEFI Forum
  - **SEC** - Security Phase, first boot phase according to PI specification
  - **PEI** - Pre-EFI Initialization
  - **DXE** - Driver Execution Environment
  - **BDS** - Boot Device Selection
- **TPM** - Trusted Platform Module international standard for secure cryptoprocessor

S-CRTM



- Saying "Secure Boot" typically means all security technologies in red
- In a reasonably secure world:
  - S-CRTM would implementation would be open (LibreBMC? lpnTPM?)
  - System ROM would contain Open Source Firmware without binary blobs
  - Bootloader/OS/Hypervisor would use standardized way for taking over and continuing chain of trust

How is this related to VMs?

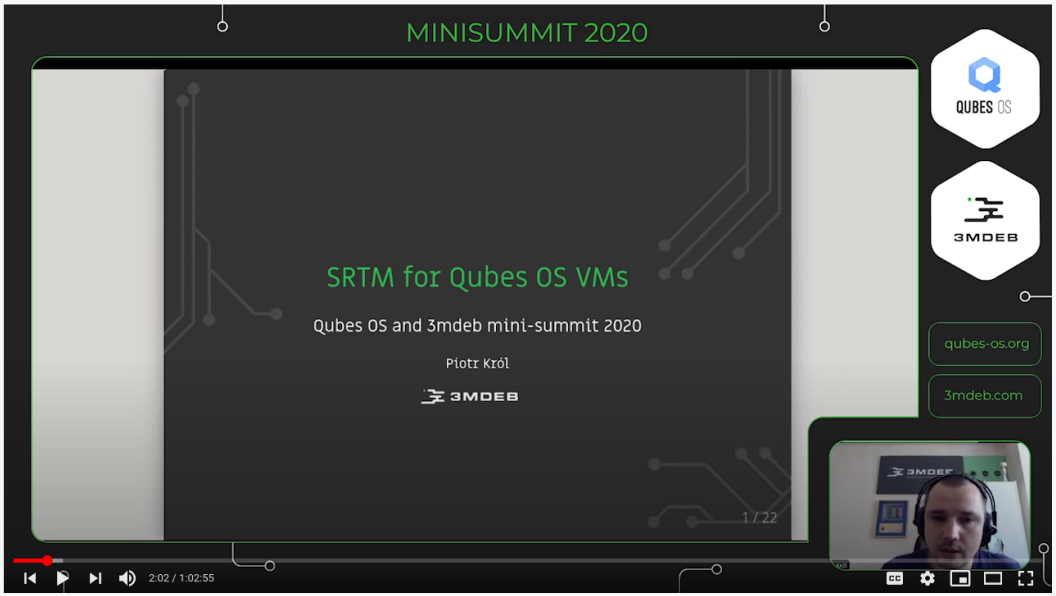


- Without chain of trust rooted in Static Root of Trust for Measurement and Verification both VM measured boot and verified boot adds nothing (or very little) to security properties of the system
  - this could be fixed by D-RTM and TrenchBoot project
  - it does not mean we shouldn't try to create PoC and experiment with solution that moves us towards measured and verified boot for VMs as default for our systems
- Further we discuss what happen over last year in scope of measured and verified boot

What happen in the topic over last year

- May 2020: Qubes OS mini-summit
  - SRTM for Qubes OS VMS - Piotr Król, 3mdeb
  - Discussion after the talk with Andrew and Marek
- Mar 2021: Xen Secure Boot and Lockdown WG Meeting
- May 2021: Xen Developer & Design Summit 2021
  - Enabling UEFI Secure Boot on Xen - Bobby Eshleman, Vates SAS
  - Alternative vTPM 2.0 Backend to Comply with Upcoming SVVP Changes - Igor Druzhinin, Citrix
- XCP-ng work around Secure Boot for VMs
- Trammel work around swtpm and safeboot
- Not directly involved by may affect openness of discussed solutions
  - LibreBMC
  - lpnTPM

SRTM for Qubes OS VMS



- TCG Roots of Trust and practical use cases of those for VM environment
- Design ideas and limitation of introducing TPM through QEMU features
- Introduced swtpm and potential configurations that can leverage it
- Outdated vTPM architecture and 3mdeb's dream design
- assumptions and potential future idea for TPM usage in Qubes OS

Key takeaways

- Building features in correct order is key to accomplish anything meaningful
  - however it is important to proceed with PoC and testing, since problem is complex, so we should divide and conquer
- There are significant difference how this could be handled by various types of VMs: PV, HVM, PVHVM, Xen/Arm, PVH
  - PVH seem to be the trend so considerations related to QEMU become obsolete and there is need to have TPM PV driver
- Look into "Virtualized Trusted Platform Architecture Specification"

TCG VPWG Architecture

- 2011 spec covers architecture, terminology, deployment models and properties that virtualised trusted computing platforms (vPlatform) are expected to offer
  - great food for thought for anyone working on vTPMs and virtual root of trusts
- Use cases
  - Creating a New vPlatform (e.g. OS requires certain TC properties)
  - Instantiating a Previously Executed vPlatform (e.g. after migration or shutdown)
  - vPlatform Operation (e.g. any application using TPM in virtualized OS)
  - Hot Stand-by (e.g. when high availability trusted vPlatfroms are needed)
  - vPlatform Upgrade
  - vPlatform Migration (e.g. migration based on attestation result)

https://trustedcomputinggroup.org/wp-content/uploads/TCG_VPWG_Architecture_V1-0_R0-26_FINAL.pdf

TCG VPWG Architecture

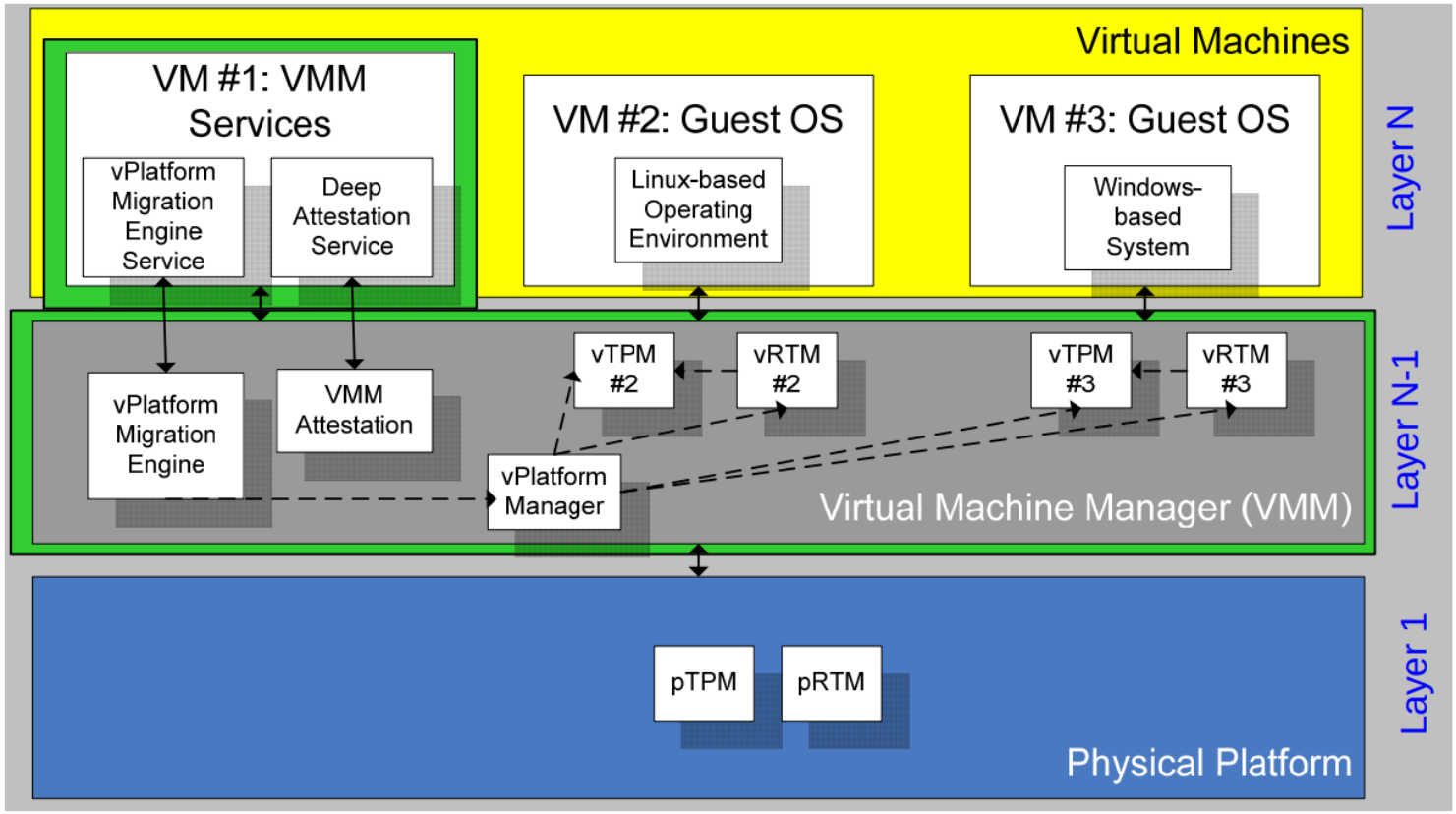


Diagram 2: Three layer architecture with privileged VM

TCG VPWG Architecture

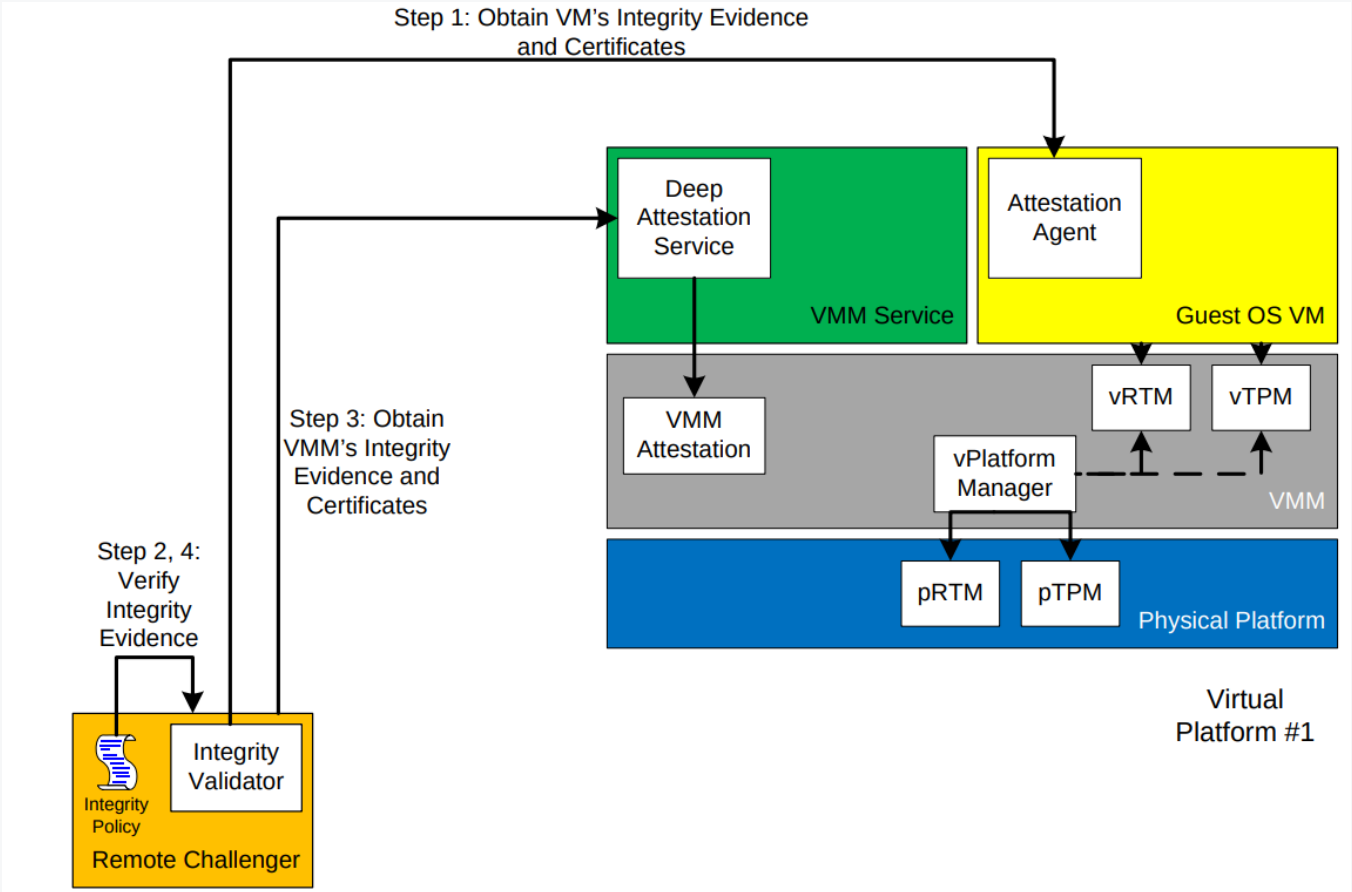
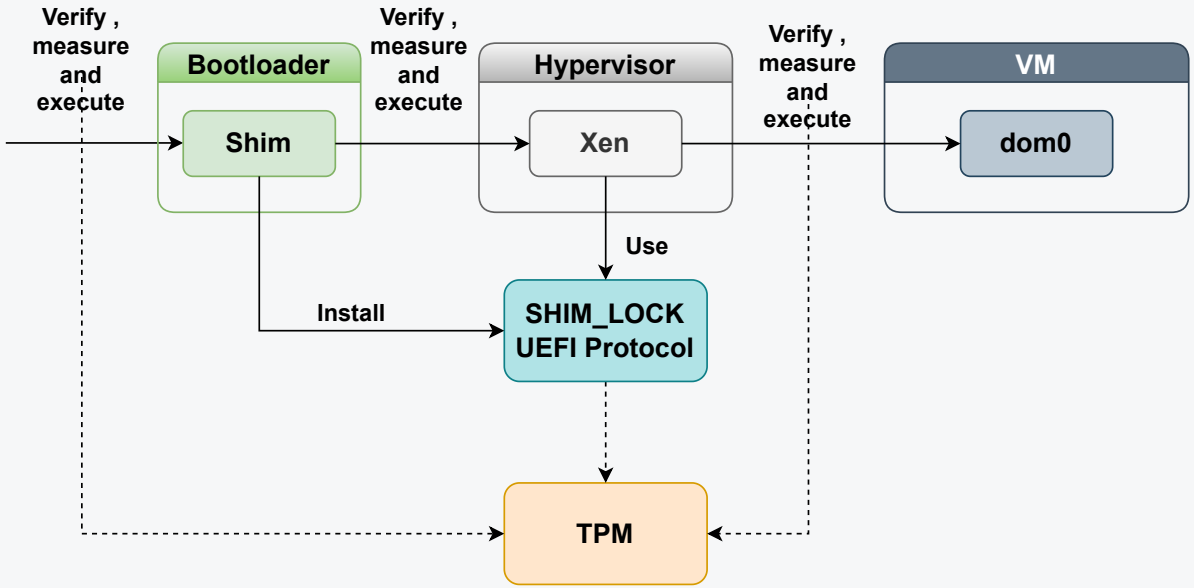


Diagram 5: Component Interactions during an Attestation of a Virtual Machine

Enabling UEFI Secure Boot on Xen

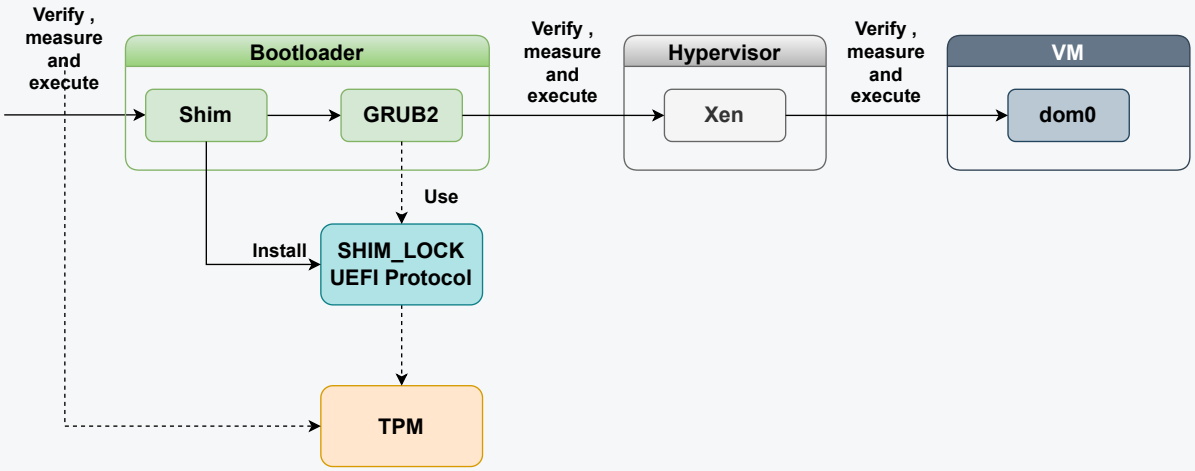
- Presentation discussed
  - what is Secure Boot and how it works
  - key hierarchy for UEFI Secure Boot
  - chain of trust in shim-present system and how it can be used on Xen systems
  - various configurations in which Xen can leverage UEFI Secure Boot and its limitations
- Side note: Microsoft CA as signing authority
  - according to rumors, they took that position because nobody else in UEFI Forum wanted or had capability

Potential configurations



- Key problem with this configuration is that it works only with xen.efi PE/COFF, but not with multiboot1/2, what means maintenance of multiple Xen build targets and lack of legacy support
- This configuration works for unified (including cmdline, params etc.) and not-unified xen.efi

Potential configurations



- GRUB2 currently support GnuPG detached signatures and Shim compatible verifier
  - it means this configuration give ability to support different then UEFI chain of trust schemes
- Deniel Kiper talk from PSEC 2018 discuss UEFI Secure Boot, Xen and Shim
  - this patches were picked by XCP-ng team
  - there is ongoing discussion on xen-devel

Key takeaway

- From Bobby email to xen-devel, the goal is to have Xen binary that can be:
  - Verifiable with shim (PE/COFF)
  - booted on BIOS platforms via grub2
  - booted on EFI platforms via grub2 or EFI loader
- Both Linux and Xen suffer from the same issue, which is potential of using some software features to subvert at runtime security properties provided by UEFI Secure Boot - Linux addressing that through Linux Kernel Lockdown
- This problem means locking down some features and that's why Xen Lockdown is the thing which have to be addressed first to get value from Secure Boot
  - Linux Kernel Lockdown would be used as model
  - unfortunately it would not be enough, because of the dom0 superpowers

https://man7.org/linux/man-pages/man7/kernel_lockdown.7.html

Key takeaway

- From OSF, Qubes OS and Xen project perspective we should think how to not break LVFS/fwupd and other firmware update methods, while implementing Xen Lockdown
    - we know that assuming proprietary/IBV SMM-based updates are the solution is not the way to go

vTPM 2.0 and SVVP

- Recent Microsoft Server Virtualization Validation Program seem to put pressure on hypervisors providers in light of TPM2.0 support
  - this may revive old vTPM Xen architecture in long run
  - it make TCG VPWG Architecture and its use cases relevant
  - it cloud be another use for swtpm in Xen
- Certification will start very soon (H1'21)

XCP-ng UEFI Secure Boot for VMs

- Current state of the work related to Secure Boot is on Github: https://github.com/beshleman/xcp-host-secure-boot
- VMs using OVMF already support Secure Boot
- based on compilation and provisioning process all required UEFI databases can be populated according to user or admin needs
- One of the discussion threads: https://github.com/xcp-ng/xcp/issues/294
- Kudos to XCP-ng team for picking up the topic

Secure Boot and Lockdown WG

- Meeting was initiated by Olivier from XCP-ng team and happen 29th March 2021 and gathered quite big crowd
- Key accomplishment was gathering of requirements for Xen Lockdown
- Verified boot chain
- Linux Lockdown - basic stuff seem to work with Qubes OS but more testing is needed
- Xen Lockdown work items
    - Live patching
    - kexec
    - /priv/cmd
    - PCI pass-through
    - QEMU
    - command line

- Trammel and couple other people work with swtpm, kexec and safeboot
- They proved it to work in couple scenarios
  - coreboot+heads testing in QEMU
  - OVMF testing in QEMU
  - attestation for both above cases
  - kexec Windows
  - safeboot support
- Overall this effort proves swtpm in various use cases discussed in this presentation

Challenges

- Wide spread and it will definitely take time to enable all necessary component
- Luckily presented concepts are already on corporate agenda and there is some pressure to move some of mentioned concepts forward
- verified boot (including UEFI Secure Boot) needs
    - hardware root of trust
    - correctly implemented chain of trust
    - ideally if would support provisioning and re-owning using open tools
    - verified boot for VMs does not have those properties, so it is as good as the weakest component executed before (BIOS, firmware, hypervisor, dom0)
- covering hypervisor and dom0 with verified boot is challenging
    - BIOS and firmware should already be covered by other verified boot technologies (coreboot vboot, UEFI Secure Boot)
    - Xen verified boot has similar challenges as Linux - this was covered by Linux lockdown mechanism

3MDEB

Q&A