SRTM for Qubes OS VMs

Qubes OS and 3mdeb mini-summit 2020

Piotr Król

3MDEB

Agenda

- `whoami`
- Presentation goal
- Terminology
- Practical use cases
- Where is S-RTM on real hardware?
- Qubes OS and S-RTM
- TPM support in QEMU
- swtpm
- Xen vTPMs
- Possible solutions
- Assumptions and future ideas

whoami

Piotr Król
Founder & Embedded Systems Consultant

- open-source firmware
- platform security
- trusted computing

@pietrushnic
piotr.krol@3mdeb.com
linkedin.com/in/krolpiotr
facebook.com/piotr.krol.756859

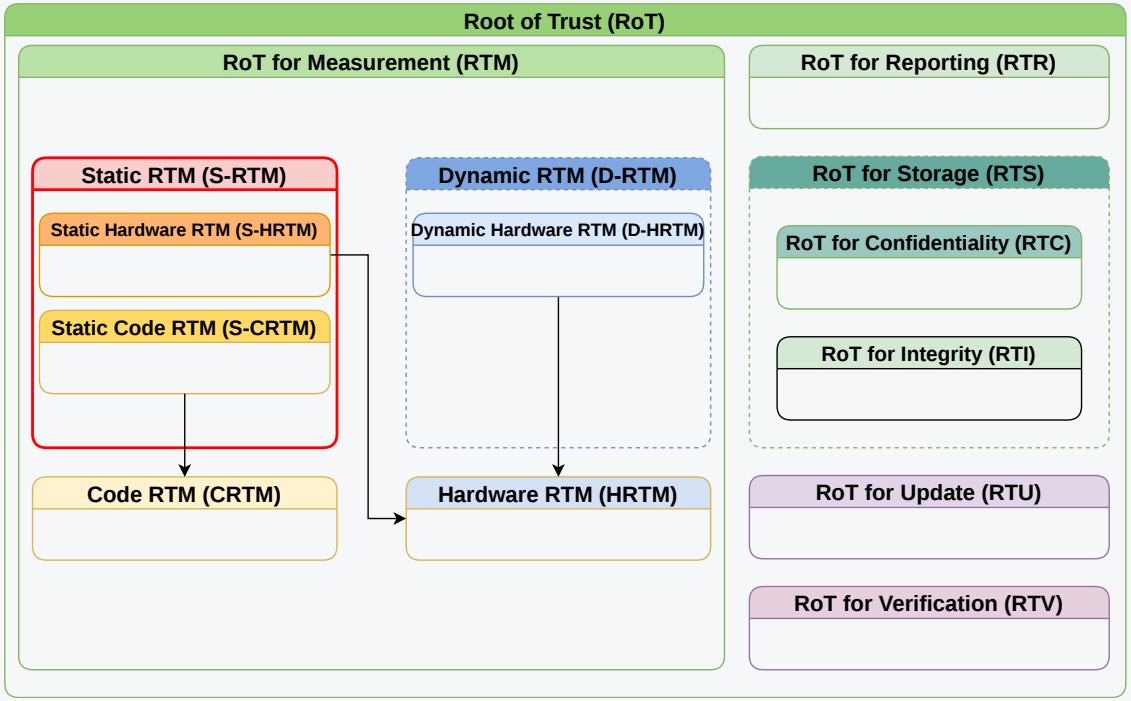
Presentation goal

To discuss feasibility and security of various S-RTM implementations for Qubes OS virtual machines

Kudos for support

- Marek Marczykowski-Górecki (Qubes OS)
- Stefan Berger (swtpm)
- Andrew Cooper (Xen)
- Daniel Smith (TrenchBoot)

Root of Trust family

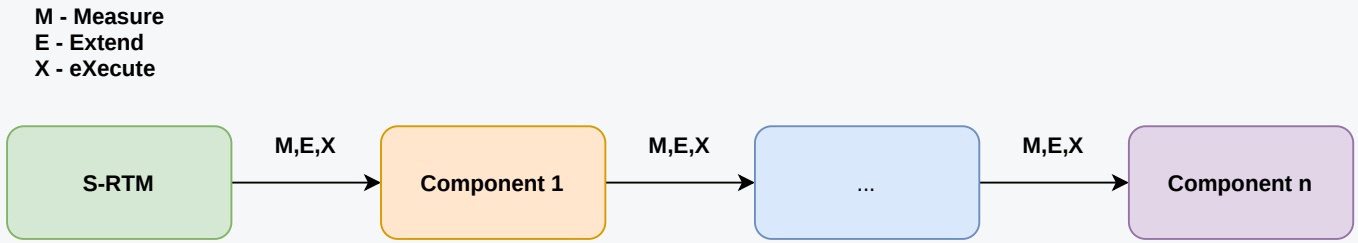


- In this presentation we will focus only on S-RTM and how to use it in virtual machines
- All those can be implemented with support of TCG complaint TPM

Terminology

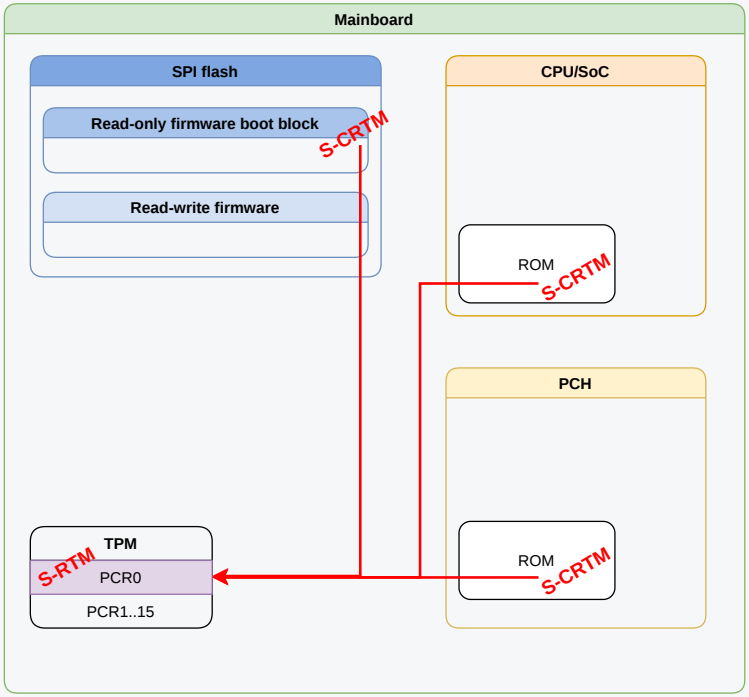
- **S-RTM** - *Static Root of Trust for Measurement*
  - *Root of Trust* because it is unconditionally trusted component, for which misbehavior cannot be detected
  - *RoT for Measurement* because it is responsible for initial integrity measurement
  - *Static* because shielded location (PCR) initialization and initial integrity measurement occurs only on platform reset
  - *Static* because it is done once for static/unchangeable components
- **S-CRTM** - *Static Code Root of Trust for Measurement*
  - TCG changed glossary and despite many places use Core, correct is Code
  - Typical example is part of firmware in read-only region of SPI flash or ROM
  - We usually call this region SEC/bootblock/BootROM or Initial Boot Block
  - On marketing level S-CRTM undistinguishable from S-HRTM

Practical use cases



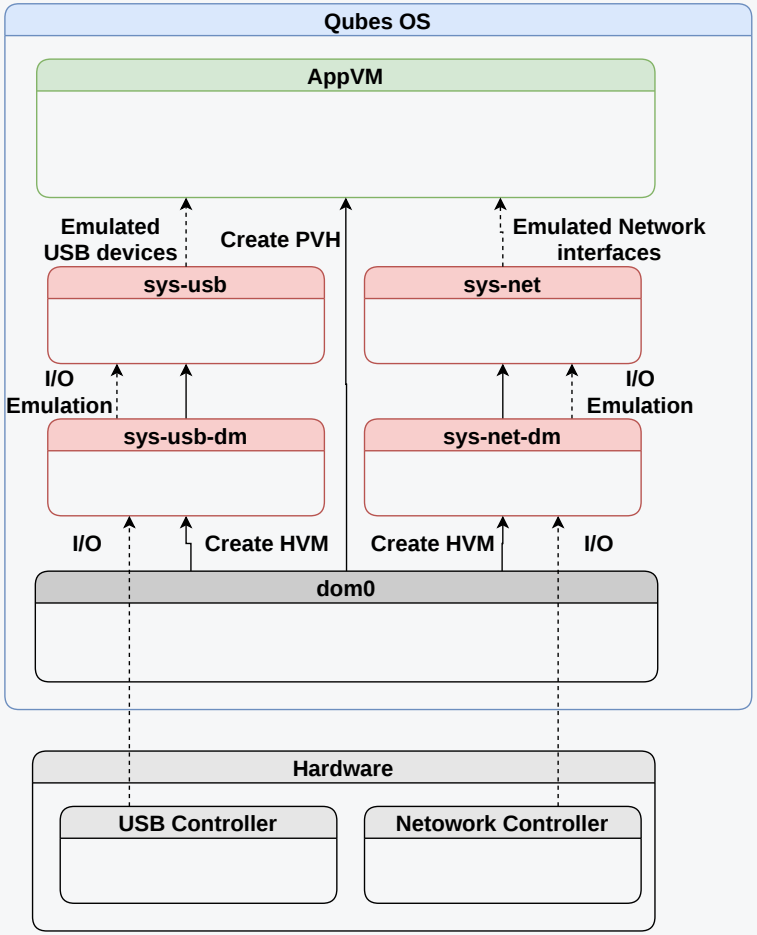
- S-RTM builds foundation for transitive trust chain which establish chain of trust e.g. during boot process (aka measured boot)
- Real value came from verification of gathered measurements (aka Attestation) or unlocking secret when measurements are correct™ (aka Sealed Storage)
- Both attestation and sealed storage may have interesting use cases in virtual machine world
    - hardened ChallengerVM that attest AppVMs measurement
    - LUKS2 encrypted disk of AppVMs that decrypts only in light of correct PCR value

Where is S-RTM on real hardware?



- After realizing where S-RTM is and how it is created on real hardware, how to move that to VMs?

Xen stub domains



- In reality most straight-forward way for enabling S-RTM and measured boot in Qubes OS VMs is adding TPM device to QEMU
- What options of enabling TPM in QEMU we have?
  - It depends on amount and types of scenarios we consider valid™
  - Qubes OS R4.0 use QEMU 3.0.0 which may limit our options for TPM, but R4.1 will use QEMU 4.2.0, which should cover most the bases
- Boot firmware:
  - Shall we limit ourselves to UEFI-aware guest OSes?
  - "we shouldn't aim to prohibit non-UEFI, but plan to start with UEFI"
  - Legacy lead to lot of issues including presence and support for TCG INT 1Ah
  - Qubes OS by default use SeaBIOS as boot firmware
- What bootloaders and operating systems should be supported?
  - Qubes OS is mostly Linux with some part of Windows VMs
- Xen version overall may also be important factor

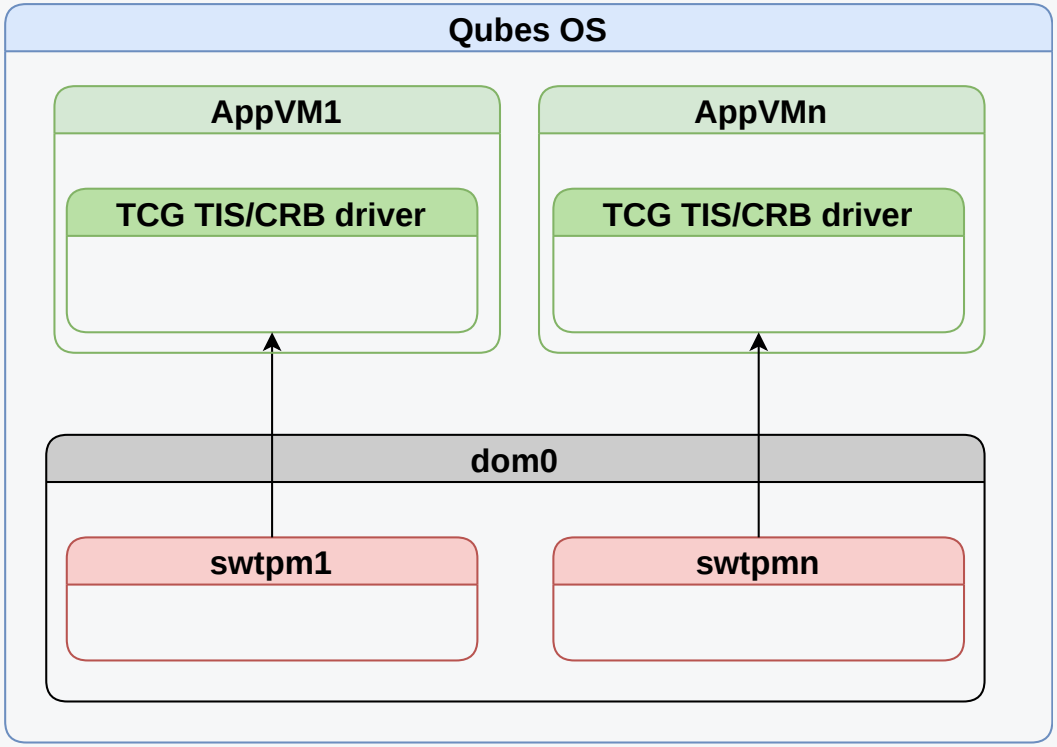
TPM support in QEMU

- TPM support consist of two pieces frontend and backend
- Frontend provide hardware interface to guest
- For x86 it can be:
  - TCG TIS (TPM Interface Specification) compatible - MMIO `0xfed40000-0xfed44fff`
  - TCG CRB (Command Response Buffer) Interface - MMIO `0xfed40000-0xfed40fff`
  - fw_cfg interface - only for Physical Presence Interface (PPI) memory, TPM version, PPI version
  - ACPI interface - only for base address obtaining, usage mode (polling vs IRQ), TCPA or adequate TPM2 table
  - ACPI PPI Interface
- Only TCG TIS and CRB use make sense in case of measured boot
- Backend provide implementation of interaction with TPM device
- There are 2 types of backends
  - pass-through - which means passing host device to VM
  - emulator device - currently only SWTPM protocol is supported

swtpm

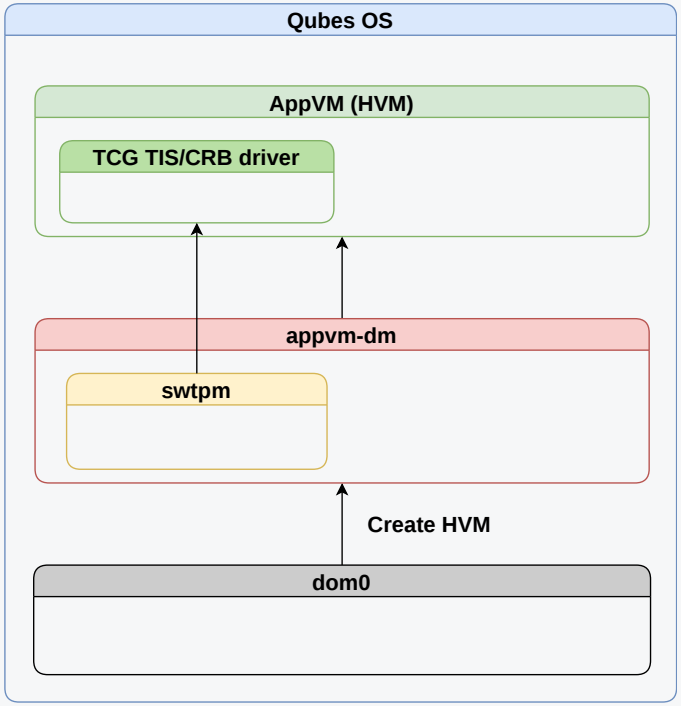
- libtpm-based TPM emulator created by Stefan Berger (IBM) in 2015
- it can use socket, character device or Linux CUSE (character devices in user space) interface for communication
- supports TPM1.2 and TPM2.0
- Threat model:
  - the same as for regular processes in hosting OS, hooking strace or gdb may reveal sensitive information
  - process running in stubdomain expose limited protocol to QEMU, which use device model to expose device to OS in VM
  - no protection against malicious hypervisor or administrator
  - using hardware features like IOMMU, SGX or SEM/SEV
- Adoption:
  - Packaged by most popular distros
  - Keylime for testing
  - OpenStack Nova
- Probably most mature OSS TPM emulation environment
- There are other TM emulators, but `swtpm` is the only integrated with QEMU

swtpm in dom0



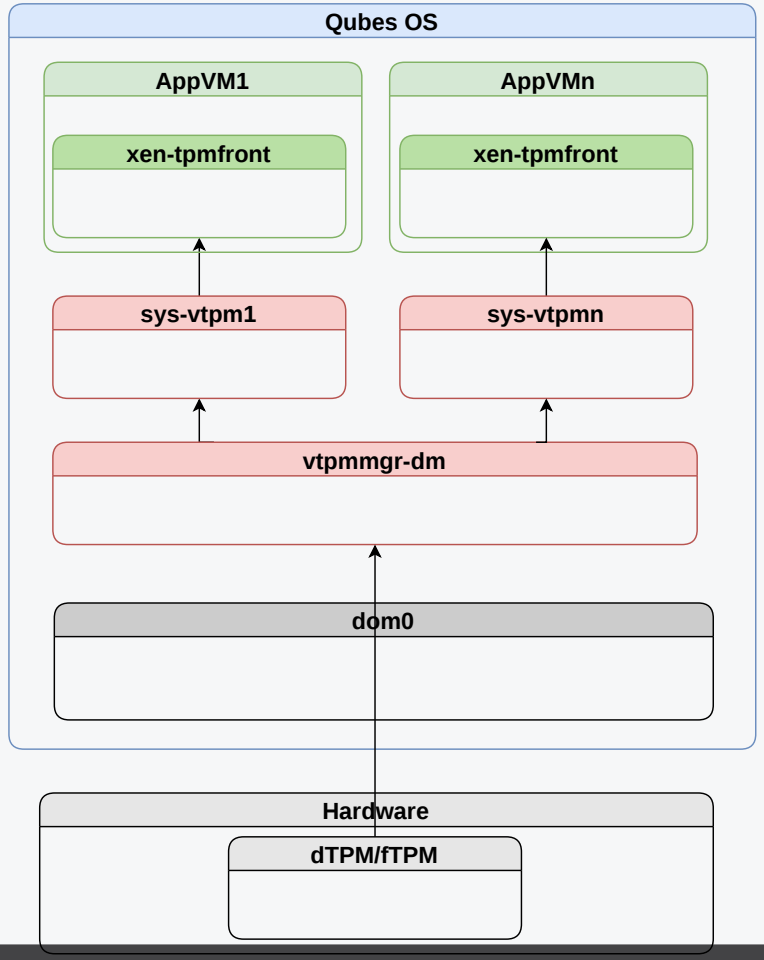
- dom0 attack vector possibly increased by `swtpm` vulnerabilities
- there is need for some mechanism managing `swtpms`

swtpm in stubdom



- Support only in HVM
- Key problem at this point is that stubdom use `-machine xenfv` which cause QEMU TPM device not being exposed to firmware and OS

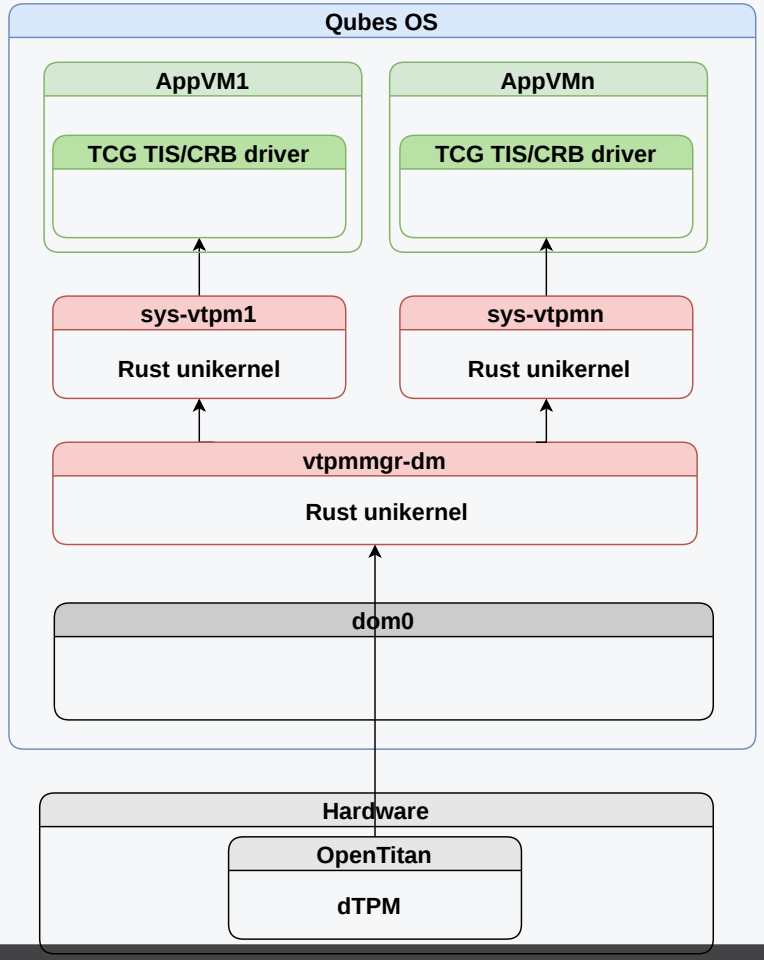
Xen vTPMs



Xen vTPMs

- Created by NSA and John Hopkins University Applied Physics Laboratory around 2010 based on previous work made by Intel and IBM
- Based on source code archaeology vTPM design for Xen was created before `swtpm`, documents refer to Berlios TPM Emulator from ETH Zurich which supports only TPM 1.2
- Natural evolution of Xen vTPM concept should use `swtpm`
- There are no public signs of using that design recently most probably because of its complexity
    - some commercial products offer vTPMs 2.0
- The goal of vTPM design was extending the chain of trust rooted in the hardware TPM to virtual machines in Xen
- `vtpmmgr` claim to support theoretically more then 20k vTPMs
    - it securely stores encryption keys of `sys-vtpms`
    - it provides a single controlled path to access physical TPM
    - it provides evidence of current configuration via TPM Quote
    - its data is secured by physical TPM seal operation
- Whole design seem to be outdated and complex

3mdeb's dream design



Threat model

- not everyone needs the same countermeasure, so either we assume recommended/feasible paranoia level or we implement environment flexible enough to satisfy various needs
- first approach seems to be better as first step, because of that we should limit our considerations to most popular solutions and assume some things

Assumptions

- Legacy boot should be avoided to make S-RTM for VMs economically feasible
  - Legacy means INT 1Ah interface implemented in BIOS and this is not well supported across the board
  - Qubes OS should consider switching to OVMF as default firmware - this increase VM TCB and maybe against Qubes OS policy(?)
- Decision between vTPM and simpler (swtpm somewhere) architecture should be made
  - it seems that vTPM consider more paranoid threat model, what probably implies its selection as target implementation
  - maybe discussion should focus on improving vTPM model?
- Limited number of OSes should be supported
  - Linux
  - Windows
- Memory consumption and complexity
  - unikernels satisfy memory consumption
- Reasonable target for S-RTM support could be Qubes OS R4.2

Future ideas and discussion

- Availability of reasonably trusted TPM to VM enables additional use cases:
  - **VM identity** - this enables various use cases including authentication, policy enforcement and remediation
  - **Shielded location for secret** - public key certificate protection, VPN credentials storage
  - **PKCS#11** - standard and programming interface which can use TPM as smart card opening another universe of use cases (e.g. payment, telecom)
  - all of those may sound like `vault-vm` or `tpm-vm`
- coreboot as default firmware for VMs
  - why? Because we consider it simpler then UEFI/edk2
- Qubes OS should have ability to choose, audit and attest VM boot firmware - it is not so easy right now
- Support for 802.1AR: Secure Device Identity which simplifies network devices authentication
- Carefully read TCG Virtualized Platform Architecture Specification

Discussion

- Interesting point is that change to image/boot firmware can be introduced only in dom0, what implies that dom0 is in TCB and should be measured ergo should be stateless
- Existence of TCG Virtualized Platform Architecture Specification indicates that S-RTM topic was considered by gov and industry leaders, but maybe whole approach was too complex to make it wide spread solution:w

Q&A