D-RTM for Qubes OS VMs

Qubes OS and 3mdeb mini-summit 2020

Piotr Król

3MDEB

Agenda

- Intro and goals
- Terminology
- Flicker
- Platform relaunch
- Virtual Machine Inspection and D-RTM
- vTPM and D-RTM
- Network booted vDLME
- Visual trust level indicator for VMs
- Future ideas
- Discussion

CC BY | Piotr Król

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

CC BY | Piotr Król

Presentation goal

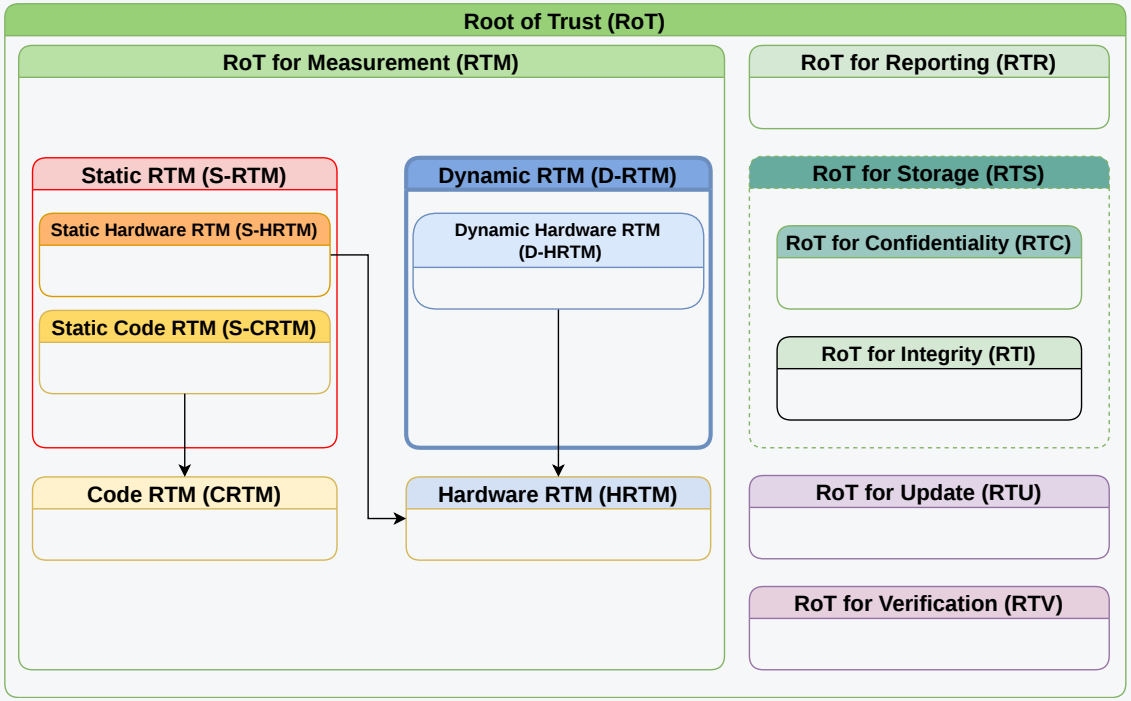
To discuss value and usage models of D-RTM implementation in Qubes OS

Kudos for support

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

CC BY | Piotr Król

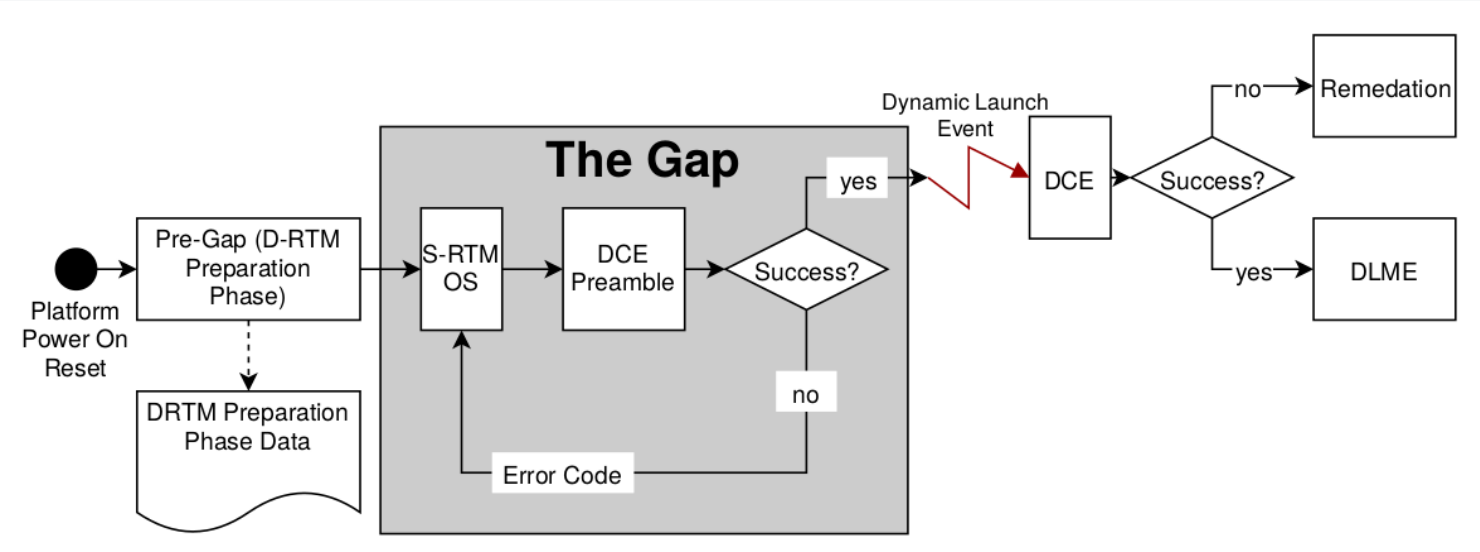
Root of Trust family



- 2 most well known implementations use special CPU instruction to trigger D-RTM (aka Dynamic Launch Event): Intel `SENTER` and AMD `SKINIT`
- D-RTM is very different from S-RTM in way it establish RTM

CC BY | Piotr Król

Terminology

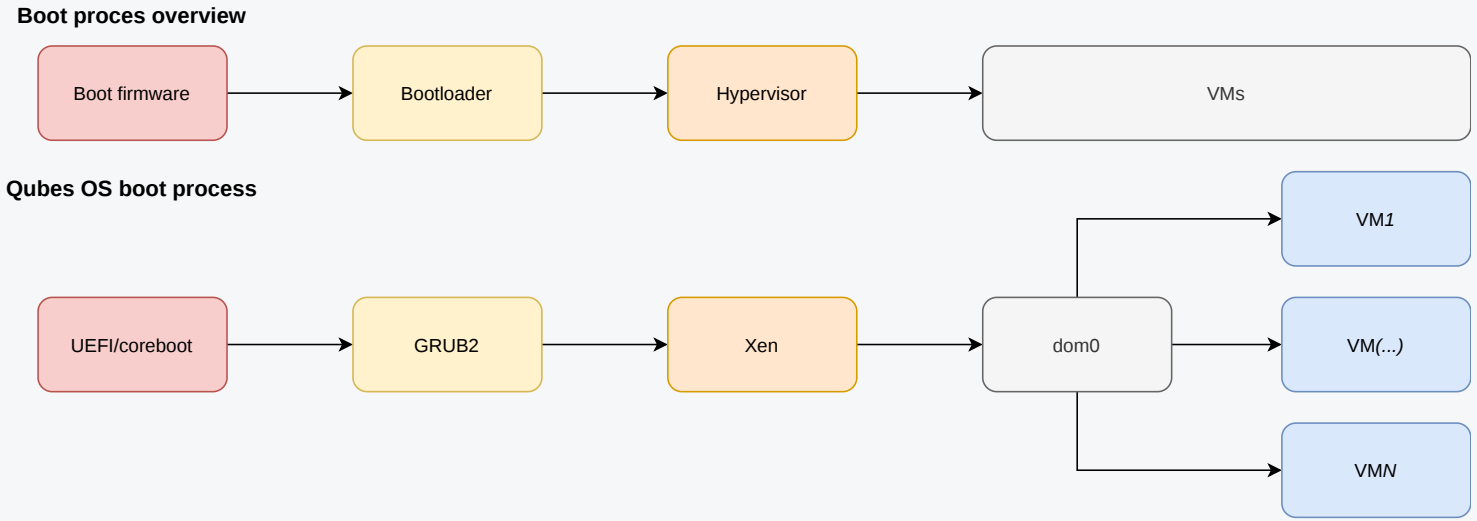


- DRTM start when Dynamic Launch event call executes
- DL Event controls PCRs 17-22, those are initialized with value -1
- DL Event change PCRs value to 0 and immediately extends with DCE hash
- Any attempt to reset TPM will set PCR[17] to -1 (TPM reset attack immunity)

TCG D-RTM Architecture v1.0.0

CC BY | Piotr Król

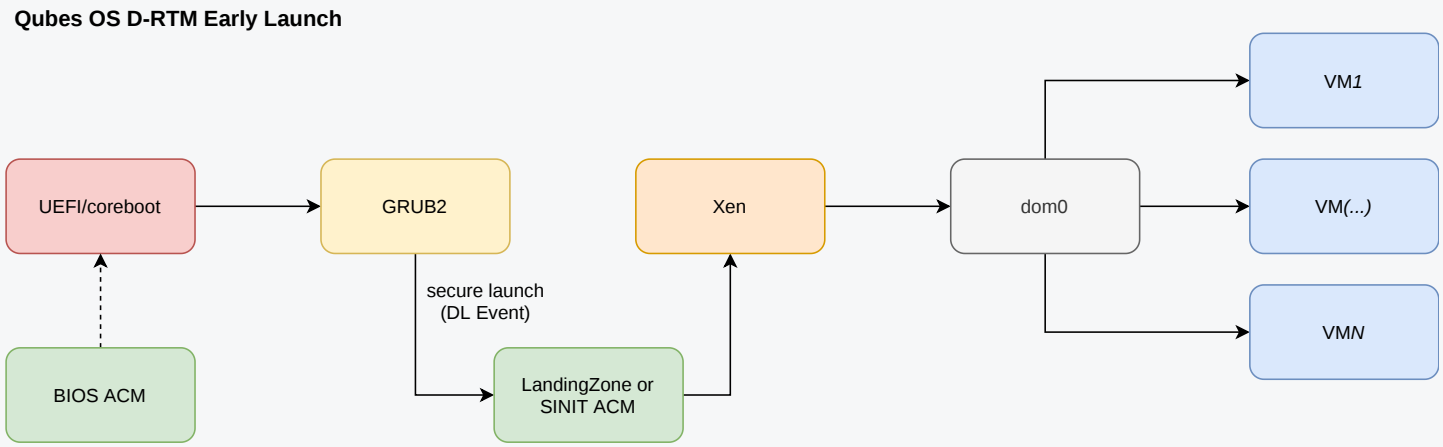
Boot process for Qubes OS



- This is just general overview of boot process to build background for further slides
- It applies to many other projects using virtualization technology

CC BY | Piotr Król

Early Launch scenario

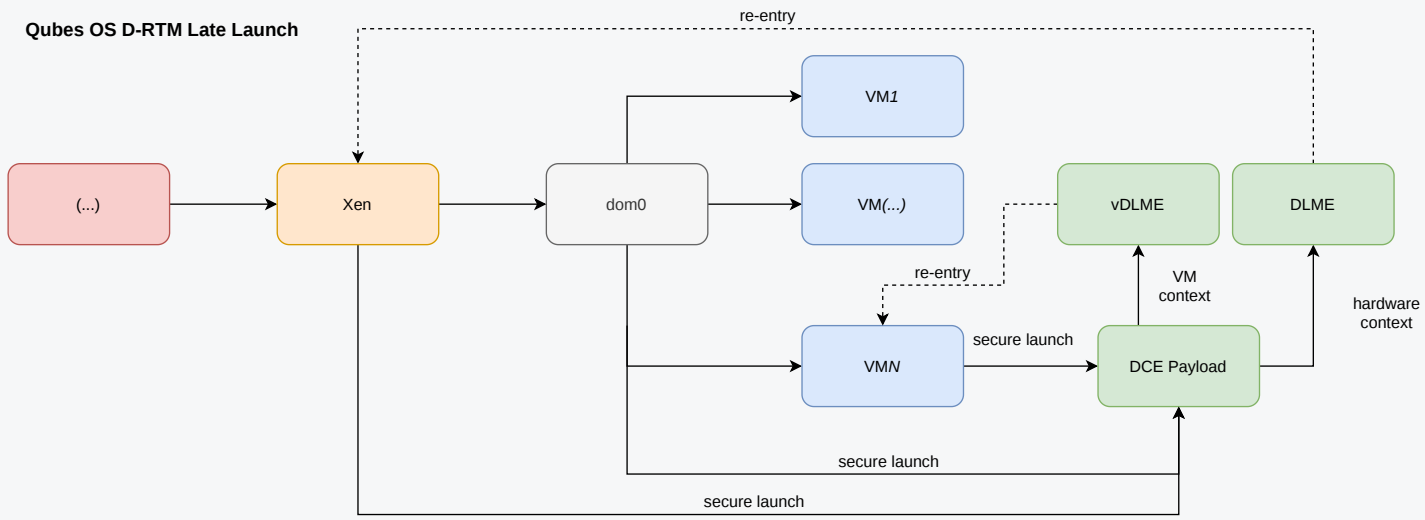


- **Early Launch** happens once each boot
- What is the value?
  - in STM-enabled configurations we can remove UEFI/coreboot from TCB
  - we can avoid complexity of S-RTM (NDAs, convoluted specs and manufacturing process)
  - we can leverage PCRs 17-22 for secret unsealing or attestation

STM - SMM Transfer Monitor

CC BY | Piotr Król

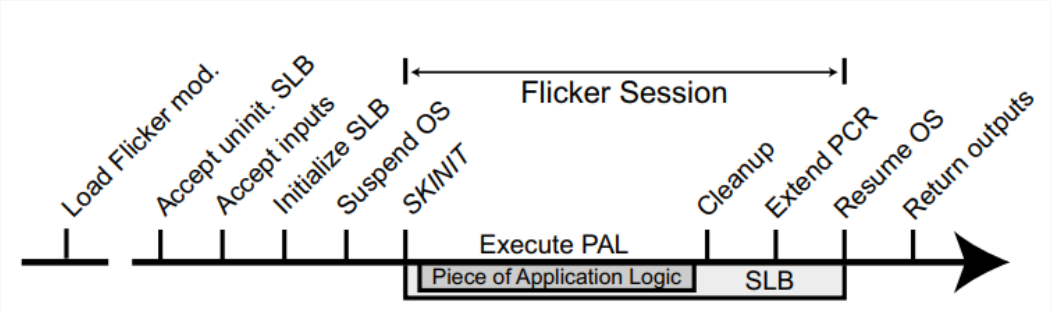
Late Launch scenario



- **Late Launch** may happen multiple times during runtime
- Way more complex case, but give lot of flexibility
- What is the value?
  - depends on DCE (D-RTM Configuration Environment) Payload
  - D-RTM provides on-demand secure execution environment
  - we will discuss further in the presentation

CC BY | Piotr Król

Flicker

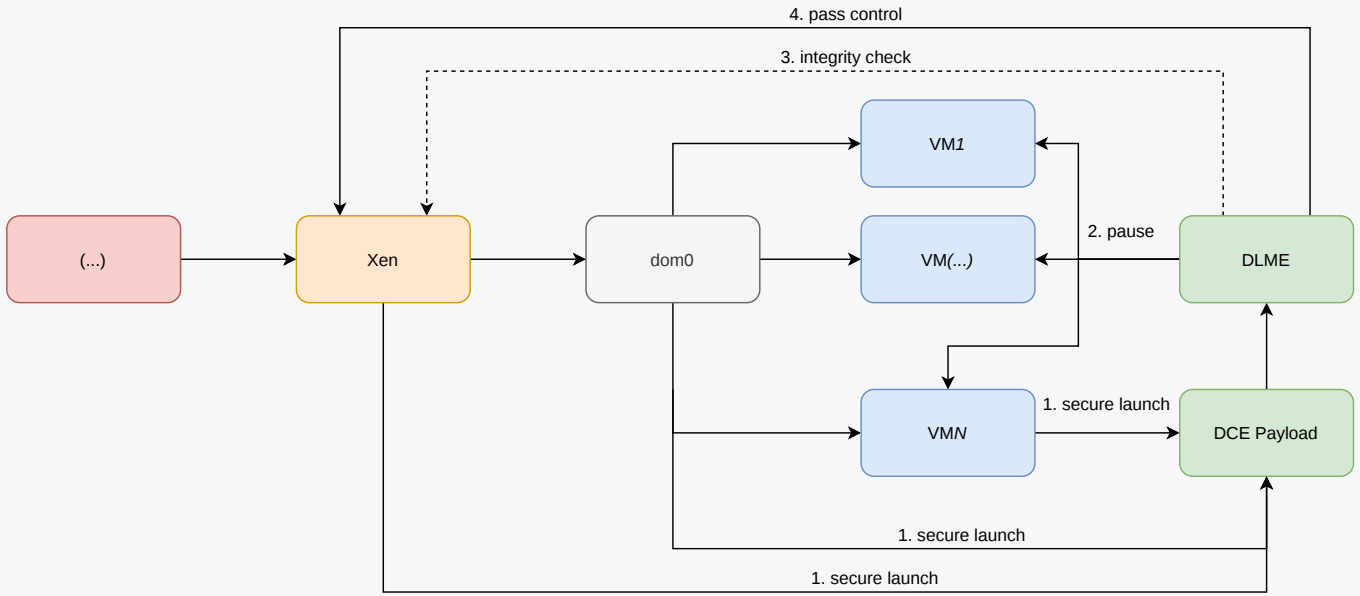


- In 2008 Jonathan M. McCune, Bryan Parno, Adrian Perrig published v0.1 version of Flicker, a technique for executing application code as DCE Payload
- Their papers provide couple interesting use cases:
  - remote rootkit detector (hash of: kernel text segment, system call table and loaded modules)
  - attested results for distributed applications
  - ssh password protection against malicious server
  - private key secure storage

https://web.archive.org/web/20160323022110/https://sparrow.ece.cmu.edu/group/flicker.html

CC BY | Piotr Król

Platform relaunch



- Since Xen can control `VMEXIT_{SKINIT,SENTER}` we have to reestablish trust in hypervisor before performing any further system integrity checks
- Otherwise Xen could trap our call or fake its results

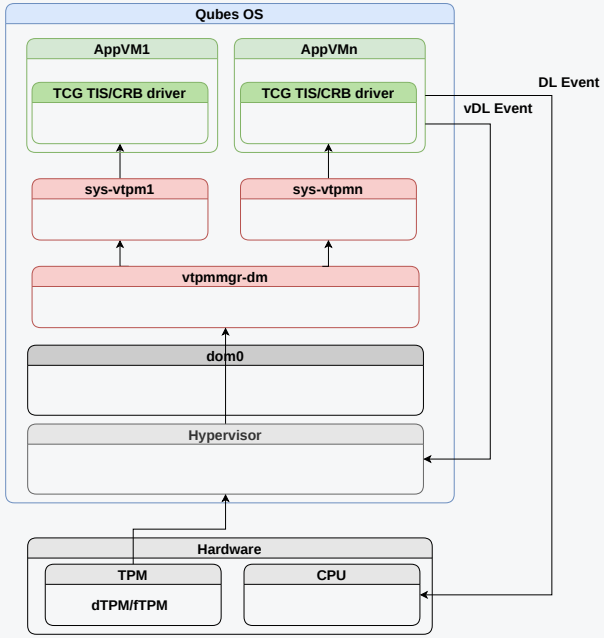
CC BY | Piotr Król

VMI

- Virtual Machine Introspection (VMI) is a technique of runtime state, what can be helpful in debugging or forensics analysis
- Use VMI for VM process table (or other running system properties) attestation:
    - critical system component periodic audits/relaunch (e.g. vault, gpg, sys-vtpm)
    - rootkit and malware detection
    - checks before performing administrative tasks
- VMI when connected with secure execution environment provided by D-RTM can bring additional use cases, which will be described later

CC BY | Piotr Król

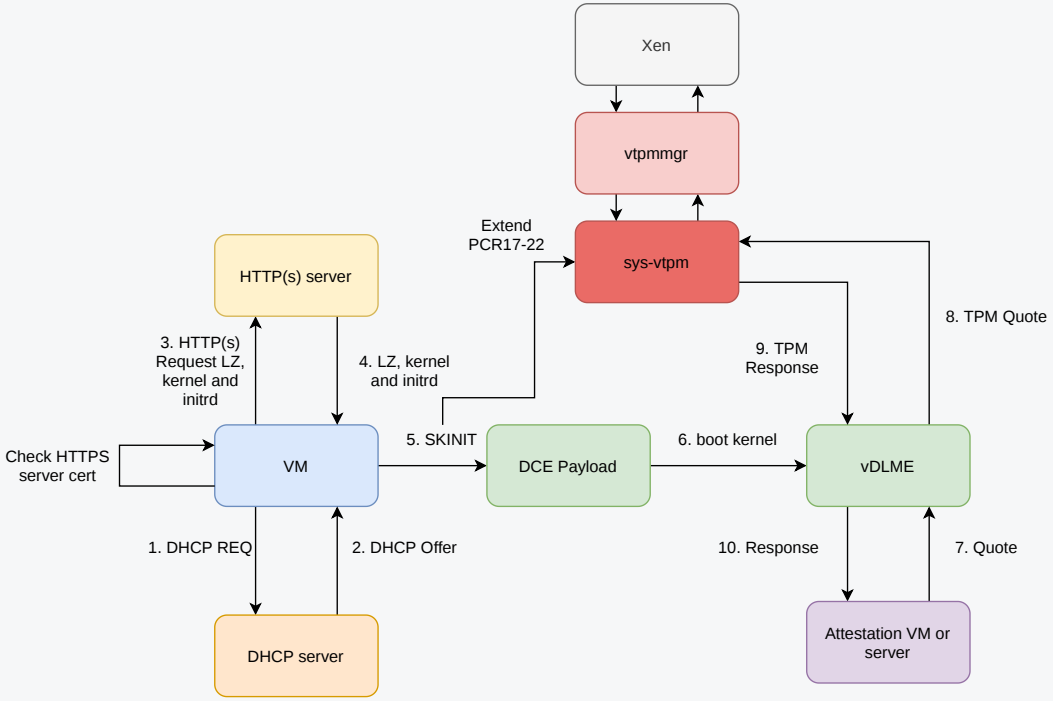
vTPM



- Any attestation of sealed secrets when using D-RTM would require TPM support for VMs
- In some cases also DL Event instruction emulation would make sense to avoid delay which real SKINIT/SENTER may cause

CC BY | Piotr Król

Network booted vDLME



- SKINIT (or any other DL Event) instruction is emulated by hypervisor but from VM perspective it looks like Late Launch

https://blog.3mdeb.com/2020/2020-06-01-ipxe_lz_support/

CC BY | Piotr Król

Visual trust level indicator for VMs

- Apply VM colors according to trust level established based on VM process table attestation
- various policies possible depending on threat model
  - if all processes are known and measured then give black
  - gradually go down to red
- early launch - color assigned at boot, also S-RTM can be used
- late launch - color change while system is running

CC BY | Piotr Król

Future ideas

- Trusted system backups and migration
  - run critical system actions only in vDLME/DLME
- Trusted firmware update
  - running critical piece of fwupd as DCE Payload or in trusted VM
- Dynamic RPC policy
  - let give ability to run some workloads only in VM that meet required trust level
  - pools of VMs meeting certain policy may change over time
  - most probably would involve developing Qubes RPC extension
- Secure storage
  - it is possible to seal secret in TPM and make it available only to certain DCE Payload
  - this may help in implementing per-VM password manager, most probably simpler and faster solution then vault VM

CC BY | Piotr Król

Discussion

- It looks like even basic Late Launch implementation to be secure requires a lot of work
- How to solve attestation problem in Qubes OS? We don't want rely on network connectivity.

CC BY | Piotr Król

3MDEB

Q&A

CC BY | Piotr Król