Qubes on modern AMD platform

Qubes OS and 3mdeb minisummit 2020

Michał Żygowski

Agenda

- Introduction
- Qubes OS on Supermicro M11SDV-4C-LN4F
- Modern AMD security features
- Secure Encrypted Virtualisation (SEV) and Qubes OS
- Current status of SEV in open-source
- Qubes OS future on AMD processors
- Q&A

Introduction

Michał Żygowski
Firmware Engineer

- @*miczyg*
- michal.zygowski@3mdeb.com
- linkedin.com/in/miczyg
- facebook.com/miczyg1395

- Braswell SoC, PC Engines and Protectli maintainer in coreboot
- interested in:
  - advanced hardware and firmware features
  - coreboot
  - security solutions

Qubes OS on Supermicro M11SDV-4C-LN4F

On the purpose of creating this presentation I have used Supermicro M11SDV-4C-LN4F board with AMD EPYC Embedded 3151 processor. This processor supports Secure Memory Encryption (SME) and Secure Encrypted Virtualization (SEV) with Encrypted State (SEV-ES) extension which will be the main topic of this presentation.

The installation of Qubes OS 4.0.1 went smoothly on this board, all qubes have been created successfully. The sys-usb qube could not be created due to the presence of USB keyboard (from BMC) and only a single xHCI controller.

DISCLAIMER: *The hardware configuration and Qubes installation used during this presentation was not intended to follow best security practices that Qubes OS offer. This board has, inter alia, a Baseboard Management Controller (BMC), which allows a remote access to the machine. It weakens the security offered by the Qubes OS.* ***Do not use Qubes OS on this machine in production environments unless you are aware of the limitations.*** *However, in the time of COVID-19 it occurred useful during the remote work.*

Qubes OS on Supermicro M11SDV-4C-LN4F

Although the initial launch of Qubes OS was successful, there were significant issues stemming from the follow two root causes:

- The xHCI USB controller broke and my USB sticks with Qubes installation image could neither be detected by BIOS nor by Dom0. Dom0 dmesg was spammed with USB errors
- Xen 4.8.4 is quite old and reported many firmware bugs and issues with APIC and interrupt vectors

Considering the above issues and the need to check the SME and SEV, I decided to move to newer Xen and Linux [1]. So I asked Marek Marczykowski-Górecki for a nightly Qubes build.

1) https://github.com/AMDESE/sev-tool#os-requirements

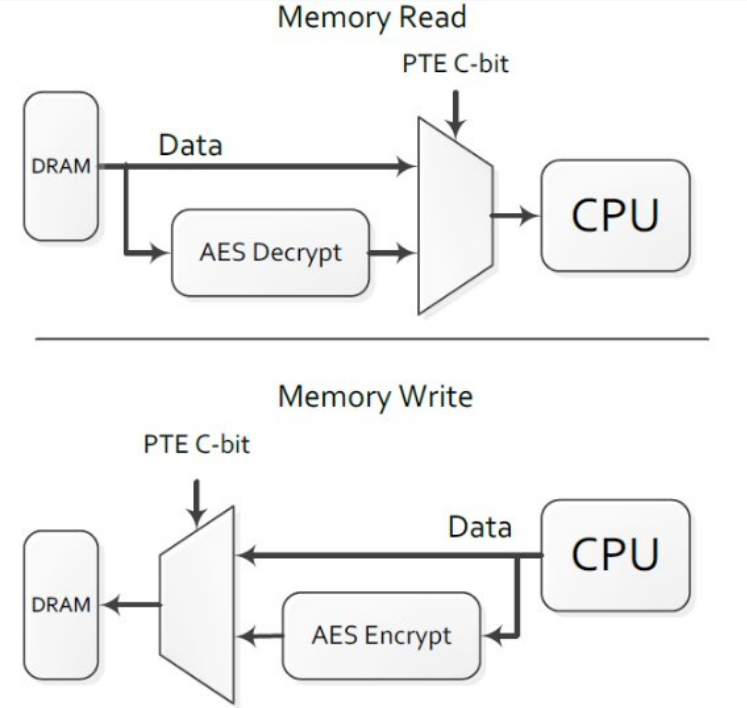
Modern AMD security features

Secure Memory Encryption (SME):

- divides into SME and TSME (Transparent SME)
- encrypts the DRAM content
- TSME is enabled at BIOS level and encrypts whole memory without any interaction from software (thus transparent to OS)

Modern AMD security features - SME

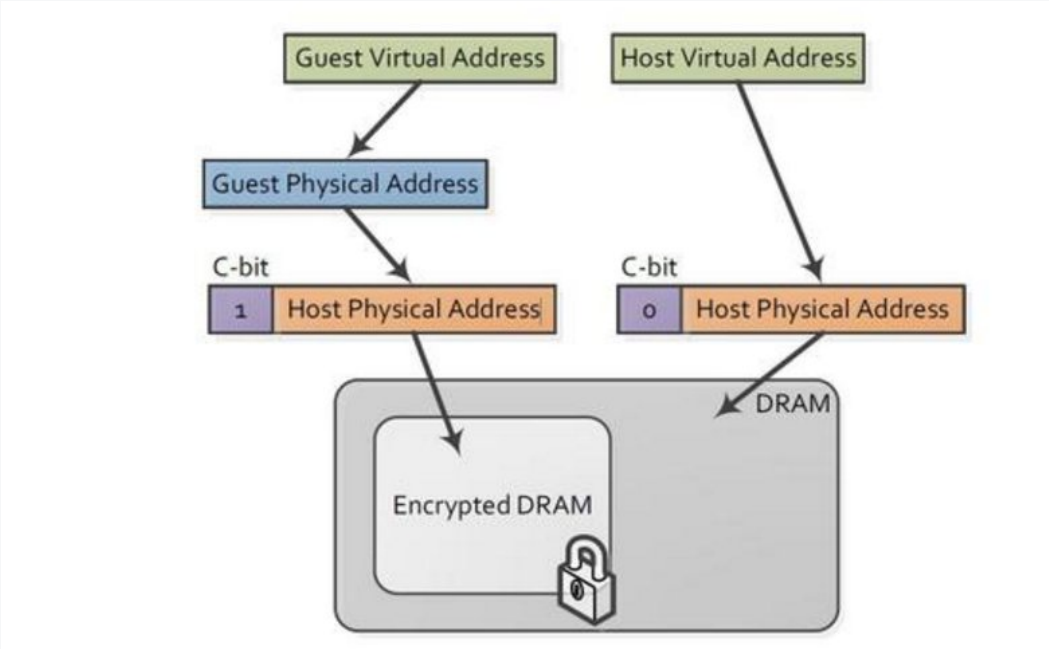
- the encryption is transparent to software and CPU operation (data read from DRAM is automatically decrypted by cryptographic engines)



Source: https://developer.amd.com/wordpress/media/2013/12/AMD_Memory_Encryption_Whitepaper_v7-Public.pdf

Modern AMD security features - SME

- SME can be enabled by operating system, which decides what is encrypted and what is not (by placing the code and data on the address space where the C-bit is set)



Source: https://developer.amd.com/wordpress/media/2013/12/AMD_Memory_Encryption_Whitepaper_v7-Public.pdf

Modern AMD security features - SEV

Secure Encrypted Virtualization (SEV):

- requires memory encryption to be available but is orthogonal to SME
- encrypts the guest VM data and code
- protects against accidental data leaks
- has two extensions:
    - SEV-ES - Secure Encrypted Virtualization - Encrypted State
    - SEV-SNP - Secure Encrypted Virtualization - Secure Nested Paging

Secure Encrypted Virtualization - Encrypted state:

- extends SEV to provide register encryption for even stricter guest protection
- created to protect from malicious hypervisors

Modern AMD security features - SEV-SNP

Secure Encrypted Virtualization - Secure Nested Paging:

- builds upon SEV-ES to provide more guest security mechanisms and reduce the trust in hypervisor
- features:
  - VM privilege levels,
  - interrupt injection restriction,
  - memory page protection from corruption,
  - data replay attack protection,
  - memory re-mapping and aliasing attacks protection

SEV and Qubes OS

While SME does not give any particular protection for Qubes OS except cold boot memory attack protection, SEV may occur useful. However, one must take following aspects into consideration:

- AMD encryption features rely on the AMD Security Processor (PSP/AMD-SP)
- AMD-SP/PSP on newer systems is tightly coupled to the memory controller, it is responsible for memory initialization and managing the memory encryption keys for SME and SEV
- AMD-SP/PSP is very similar to Intel ME , it runs closed firmware and CPU cannot properly operate without it
- SEV requires its own firmware loaded by SEV driver - yet another example of security by obscurity (BIOS and security ecosystem is already full of blobs and closed solutions)

SEV and Qubes OS

Using nightly Qubes OS 4.1 build I tried to check the SEV support status in Xen and Linux kernel.

- The build contained Xen 4.13.0 and Linux 5.4.31
- AMD EPYC 3151 should support SME, SEV and SEV-ES
- Memory Encryption was enabled in BIOS
- Despite that, the dmesg from Dom0 conatained:
  ccp 0000:06:00.2: SEV: failed to INIT error 0x8003
- Found an issue about it for exactly the same board:
  https://github.com/AMDESE/AMDSEV/issues/36

Despite trying my best I could not do anything with SEV. It is not surprising Linux don't see SEV under Xen - I would expect Xen handles features like this and hides it from dom0 (always, or if not supported) - especially ***kvm_amd.sev=1*** has no chance to affect it (because KVM requires bare kernel). Also tried enabling the memory encryption via kernel commandline with ***mem_encrypt=on***, but it has no chance running under Xen (lack C bit interpretation) Since the BIOS does not have any option to enable SEV, the only option is to tweak the kernel.

Current status of SEV in open-source

- it is currently possible to run SEV-enabled guests using open-source tools
- AMD has prepared guides how to launch SEV guests: https://github.com/AMDESE/AMDSEV
- requires libvirt >= 4.5 and qemu >= 2.12
- SUSE Linux Enterprise Server already has support for SEV
- guests must run OVMF (SeaBIOS is not supported)
- hypervisor and guest must support GHCB structures
- there are limitations about intercepting certain events by the host
- new #VC exception added that requires special consideration when writing a handler
- handling Automatic Exits (AE) and Non-Automatic Exits (NAE)

References:

- https://github.com/AMDESE/ovmf
- https://github.com/AMDESE/AMDSEV
- https://github.com/AMDESE/sev-tool
- https://developer.amd.com/sev/
- https://www.amd.com/system/files/TechDocs/24593.pdf
- https://developer.amd.com/wp-content/resources/56421.pdf

Current status of SEV in open-source

- There is still much work to do on Xen side to support SEV guests (restricted event intercepting, AE and NAE, GHCB, AMD-SP/PSP driver)
- SEV introduces new API to the firmware and SEV-SNP introduces new ABI, which needs to be implemented in Xen or Linux
- Lack of AMD's Xen maintainer which postpones implementation of new features in Xen
- Using SME or SEV hits the performance, which may not always be acceptable for all users
- SEV is available in Linux kernel for KVM guests and new qemu

Qubes OS future on AMD processors

- Encrypted qubes for most critical applications (vault, personal, work)
- Optional support to launch a qube using OVMF with SEV (SeaBIOS cannot be supported, since SEV requires PAE paging or long mode, but SeaBIOS is an 16 bit environment)
- Those who have SME option in BIOS, can utilize TSME feature in Xen and Linux right now. The memory will be encrypted by default without software intervention protecting from cold boot memory attacks.

Q&A

Thank you