Fraunhofer
AISEC

Improving Confidential Computing with seL4

A Promising Guest OS Solution

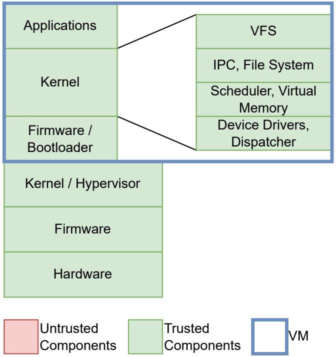
Alexander Weidinger, September 4, 2025

Motivation

- All-present "trend" of using computational resources in the cloud, often also on shared platforms, to process data
- More and more of this data is also highly sensitive (e.g., healthcare, automotive, AI, ...)
- We somehow need to protect the processed data from the cloud provider
- **Data-at-rest:** Can be solved by using disk encryption
- **Data-in-transit:** Can be solved by using secure transport protocols (e.g., TLS, VPN, ...)
- **Data-in-use:** Lesser-known and more recent - can be solved by using Confidential VMs (e.g., AMD SEV-SNP, Intel TDX, Arm CCA, RISC-V CoVE)

Cloud-Computing

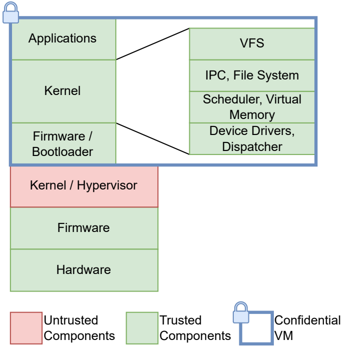
"Traditional"



- Common hardware/software stack
- Monolithic OS (i.e., Linux) with huge and complex code-base
- Hypervisor can access/manipulate memory of VM → **no** data-in-use protection

Cloud-Computing

Confidential VM



Applications

Kernel

Firmware / Bootloader

VFS

IPC, File System

Scheduler, Virtual Memory

Device Drivers, Dispatcher

Kernel / Hypervisor

Firmware

Hardware

🟥 Untrusted Components

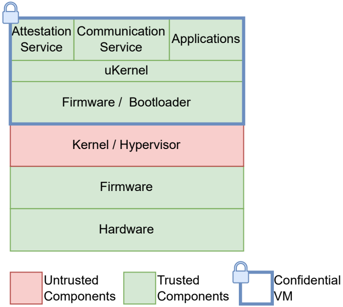
🟩 Trusted Components

Confidential VM

- Still common hardware/software stack
- Monolithic OS (i.e., Linux) with huge and complex code-base
- Hypervisor cannot access or manipulate memory of VM → **data-in-use protection**
- Device drivers still expose a large attack surface (complex, executed in kernel-space)

Cloud-Computing

Confidential VM with reduced Software-Stack



- Monolithic kernel is replaced by a uKernel (e.g., seL4)
- Only necessary functionality is implemented in kernel-space
- Allows for building minimized systems
- The kernel is formally verified and
- provides capability-based access control
- User-space drivers enable least privilege, thereby significantly reducing the attack surface

Confidential VMs

Background: AMD SEV-SNP

- Secure Encrypted Virtualization (SEV) was first introduced in 2016.
- SEV-SNP is the third-generation of AMD's SEV technology
- **SEV:** Allowed per-VM memory encryption (one key per VM)
- **SEV-ES:** Additionally encrypts CPU register state on VM exits
- **SEV-SNP:** Adds memory integrity protection (+ some additional stuff)

Confidential VMs

Concepts: AMD SEV-SNP

- SEV introduces an "en**C**rypted" Bit which marks encrypted (private) memory in guest page table entries
- SEV-ES adds a new exception type (VMM Communication Exception **#VC**) and the Guest Hypervisor Communication Block (**GHCB**), a paravirtual communication path
- SEV-SNP adds the Reverse Map Table (**RMP**), which enforces page ownership
- Additionally, SEV-SNP adds Virtual Machine Privilege Levels (VMPLs), etc.

AMD SEV-ES

NAE Example Flow

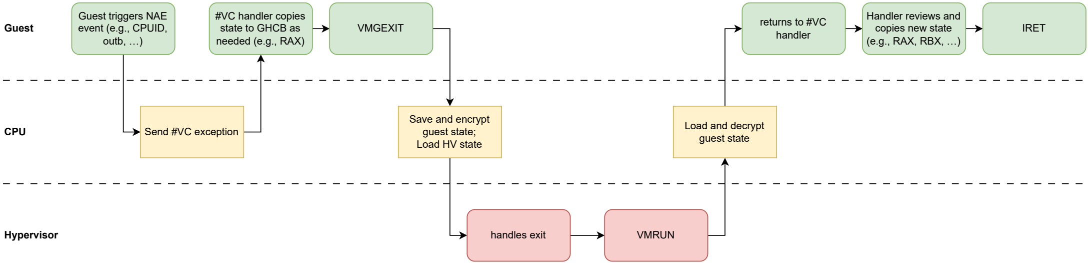

Guest
CPU
Hypervisor
Guest triggers NAE event (e.g., CPUID, outb, …)
Send #VC exception
#VC handler copies state to GHCB as needed (e.g., RAX)
VMGEXIT
Save and encrypt guest state; Load HV state
handles exit
VMRUN
Load and decrypt guest state
returns to #VC handler
Handler reviews and copies new state (e.g., RAX, RBX, …)
IRET

Figure: NAE Example Flow [Kap17]

Current State

- We made some changes to the seL4 code, mainly
- adding C-bit-aware page table setup for SNP guests,
- adding a handler for the #VC-Exception and
- implementing the necessary NAEs (i.e., *CPUID*, *IOIO* and *MSR*).
- We also had to adapt early startup code for the usage of `cpuid` pre-#VC and to register the GHCB.
- Environment: Linux Host + QEMU/KVM + OVMF/edk2 + GRUB

These kernel changes are (more or less) enough to run a first "Hello World" application. (-:

Use Cases

- Secure Cloud Block Device
- VPN-Gateway
- Key-Storage
- Cloud-TPM
- ...?

Limitations & Future Work

- Currently, our implementation is just a PoC
- No optional features are implemented, just the bare minimum (RMP is not strictly necessary for "Hello World" since firmware validates initial pages)
- Attestation is on our TODO-list
- (Formally verified) VirtIO Net/Block/... drivers
- Complex Firmware/Bootloader inside Confidential VM also provides big attack surface

Pain Points

- Debugging SEV-SNP protected VMs is possible but not implemented in QEMU yet
- We sometimes experience inconsistent behavior of SEV-SNP VMs during startup
- Stick to known-good versions of your environment! OVMF/edk2 behaves differently between versions in our experience.

Thank You!

Bibliography

David Kaplan.
Protecting vm register state with sev-es.
White paper, 46:158, 2017.

Fraunhofer
AISEC

Contact

Alexander Weidinger
Department
Secure Operating Systems
Tel. +49 89 3229986-1034
alexander.weidinger@aisec.fraunhofer.de

Fraunhofer-Institute for Applied and Integrated Security AISEC
Lichtenbergstr. 11
85748 Garching (near Munich)
Germany