SUSE

15. March 2023

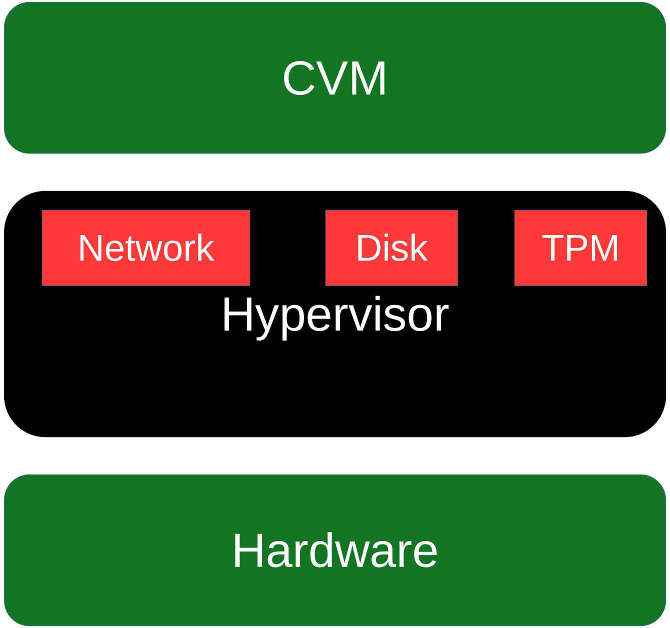
Trusted in-guest Hypervisor Services with the Secure VM Service Module

Jörg Rödel <jroedel@suse.com>

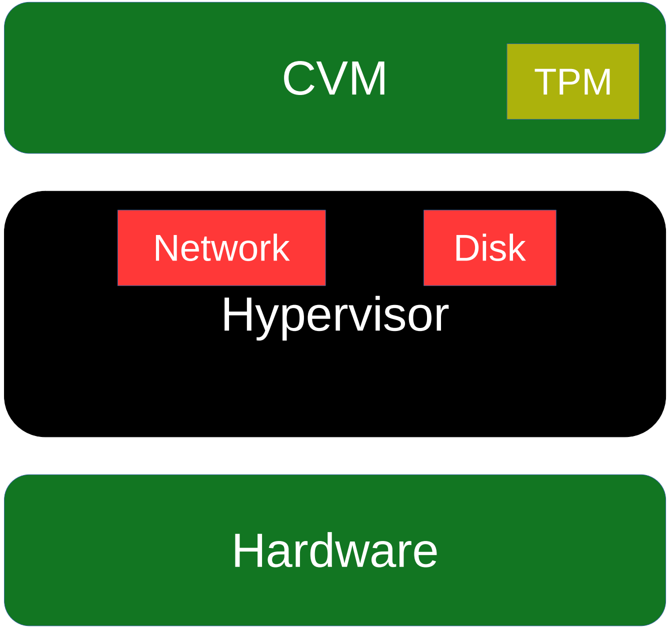
The Secure VM Service Module (SVSM)

- Builds on AMD SEV-SNP
- Uses VM Privilege Levels
- Allows in-guest device emulation
- Many use-cases: vTPM, Live migration, UEFI variable store

Secure Device Emulation



Secure Device Emulation



VM Privilege Levels

- Hardware feature available with AMD SEV-SNP
- 4 privilege levels (VMPL0-VMPL3)
- Allows memory separation within guest context
- Store data protected from the OS

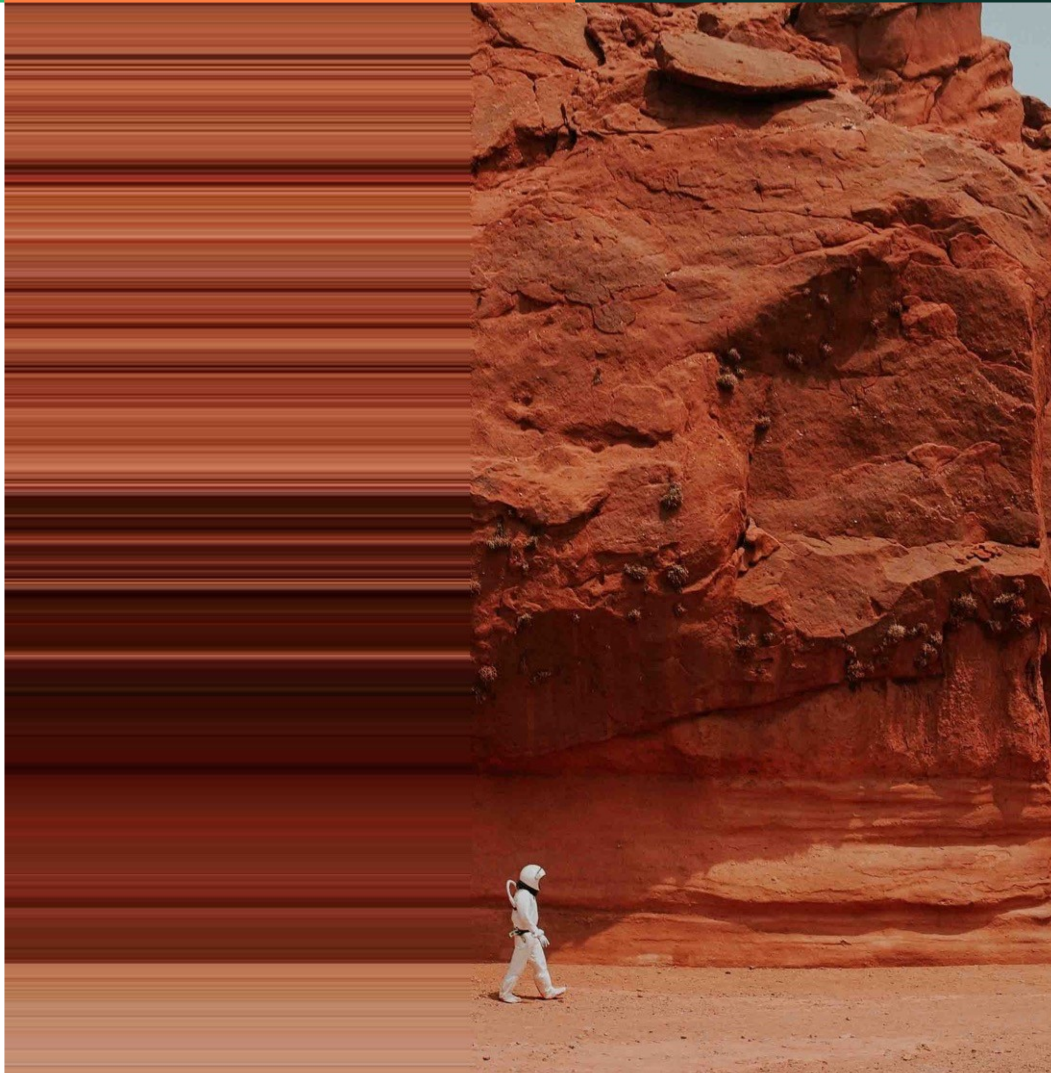
VM Privilege Levels

- Firmware (FW) and OS moved to a less privileged VMPL
- Allows a software running in VMPL0 for protected data
- Software at VMPL0 is the SVSM
- Communication with FW/OS via request protocol

VM Privilege Levels

- Some operations only allowed in VMPL0
  - PVALIDATE
  - RMPADJUST to make pages available to FW/OS
- Core protocol offers these instructions to the FW/OS

The Secure VM Service Module



Secure VM Service Module

- POC implementation in Rust by AMD (linux-svsm)
  - Comes with additional repositories for Linux host, guest and OVMF support
- Patches available for attestation and TPM emulation
  - No strong separation within SVSM yet

Secure VM Service Module

- Another implementation started by SUSE
  - Based on the Linux and OVMF patches from AMD
  - Stronger focus on isolation
  - Currently ~7700 lines of Rust code

Key Features

- PerCPU page-tables
  - Address space separation into PerCPU and shared areas
- Buddy and slab-based memory allocator (ported to linux-svsm)
- Debugging features
- Exception fixups

Key Features

- Currently boots a Linux SMP guest
- Does not use the x86-64 crate from crates.io
- Multi-stage launch process
- Can run from any guest physical address

Next Steps – CPL3 Support

- ELF loader to run binaries in CPL3
    - Additional separation within SVSM
- Needs some boilerplate code to harden entry code and exception handlers
- SYSCALL handlers and entry/exit path

Next Steps – Persistency Layer

- Allow the SVSM and its processes to safely store data
- Needed for vTPM and also UEFI variable store
- Several ideas discussed right now how to handle this

Next Steps – Launch Protocol

- Create an SVSM specific OVMF target
  - Package that together with SVSM binary
  - SVSM will unpack OVMF and launch it
- Allows to use SVSM binary as a drop-in replacement for OVMF in QEMU

Further Steps – Live Migration

- Needs an SVSM-Hypervisor communication protocol
- Handshake between source and destination SVSM

  with attestation
- In-guest page re-encryption
- Hypervisor for communication channel and dirty tracking

Vision

- Extend the SVSM into a paravisor
  - Run mostly unmodified OSes
  - Needs #VC handling in SVSM including instruction decoder
  - Be able to run Windows on KVM with SEV-SNP

SUSE

Thank you

For more information, contact SUSE at:

+1 800 796 3700 (U.S./Canada)

+49 (0)911-740 53-0 (Worldwide)

Frankenstraße 146
90461 Nuremberg
www.suse.com