Secure Live Migration of Encrypted VMs

Tobin Feldman-Fitzthum
Dov Murik
IBM Research

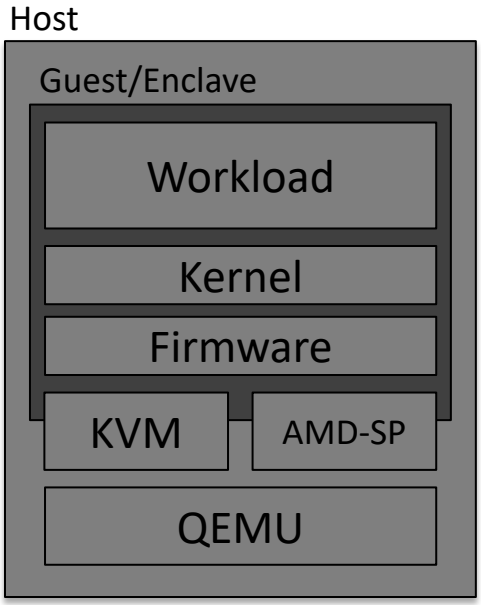
THE LINUX FOUNDATION

Migration

- Move VM from one node to another without stopping
- Hypervisor facilitates
  - Converging memory
  - Coordinating CPU state
  - Controlling execution

Confidential Computing

- Protection of data in use
- AMD SEV
  - The VM is the enclave
  - SEV, SEV-ES, SEV-SNP
- Hypervisor untrusted



SEV Live Migration

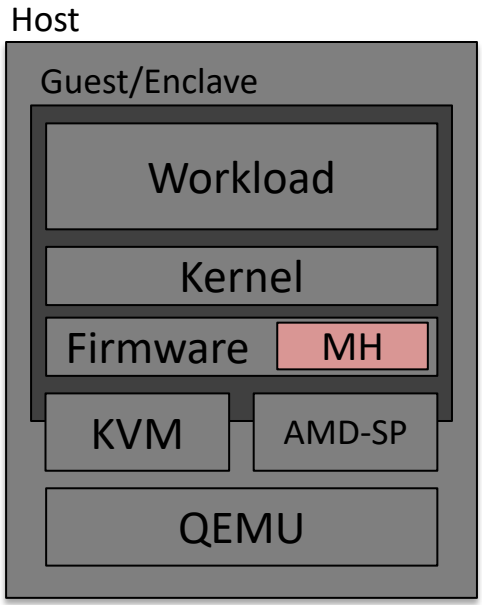
- SEV - encrypt guest memory with a key managed by the hardware
  - How does the hypervisor copy pages from source to target
    - Can't copy the ciphertext
      - won't decrypt if moved
      - and the keys aren't on the target
- SEV-ES - protect guest CPU state
  - How does the HV coordinate the CPU state between source and target?*
- SEV-SNP - integrity guarantees for memory
  - How does the HV guarantee integrity during migration
    - What if a page becomes dirty after it has been copied?

Memory Encryption

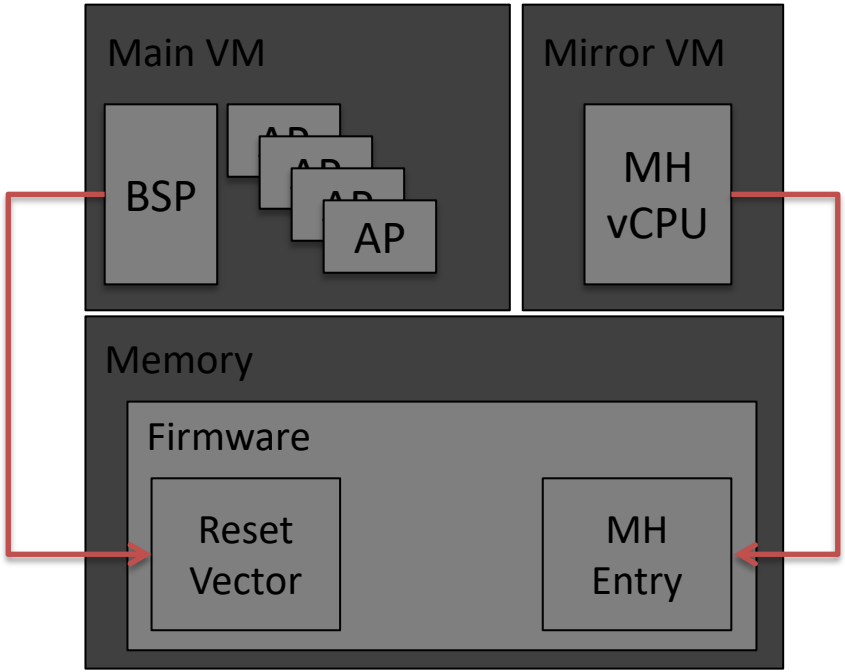
- AMD-SP does have migration support
  - Wrap pages with transport key
  - KVM Forum in 2017 and 2019
  - Insufficient throughput to copy all guest memory
- We need support from the guest
  - Migration Handler inside guest context, but not part of the workload
  - Where should it live?
  - API: export page, import page
  - Pages encrypted with shared transport key

Firmware Migration Support

- MH in firmware can be measured and at boot
  - No opaque blobs
- Minimal OS dependency
  - Migrate early in boot or with hung guest
- Auxiliary vCPU
  - Add an extra vCPU, but hide it from the OS
  - OVMF starts normally and spins up the MH on the extra vCPU
- Mirror VM
  - Create a secondary VM that shares memory and encryption context (ASID)
  - Warm boot secondary VM directly to MH



Migration Handler



- Primary VM and mirror share memory (ASID)
- Are the same to AMD-SP
- HV starts mirror vCPU from special entry point

Migration Handler

- OVMF has migration entry point
  - EIP discoverable by parsing firmware
- MH Entry trampolines to Migration Handler
- MH looks like normal DXE runtime driver
- Special mapping
  - Identity map with c-bit + shared pages at offset
- Firmware support in main VM
  - Setup the entry point

Is it safe?

- Hypervisor triggering execution inside the enclave
  - MH is measured
  - API is small
- QEMU depending on guest execution
  - QEMU can’t verify execution of MH
  - API is small
- Guest Owner verifies launch measurement of source and target
  - Transport key provided only if measurements check out
- Mirror boot process works with SEV-ES

SEV-ES Live Migration

- AMD-SP saves CPU state to encrypted memory at each VMExit
  - a handler puts CPU state needed for CPU Exit in a special buffer
- Initial CPU state is part of the launch measurement
- HV can set the initial CPU state, but the target VM will already be running
- How can we set the CPU state of a running guest?
  - Trampoline

Trampoline

- Map VMSA into the guest
  - Force an exit after memory converges
  - VMSA migrated as guest memory
  - Target MH has source CPU state in memory
- Can’t atomically resume CPU state
  - Set each register individually via trampoline
  - Delicate but possible
    - Need an intermediate page mapped in source PGT and MH PGT
    - Need trampoline for each vCPU

Trampoline

- Suspend / Resume?
  - Is this a live migration?

- SEV-SNP RMPADJUST
- No integrity protection for pages with VMSA

SEV-SNP Live Migration

- SEV(-ES) does not protect against replay attacks
- SEV-SNP guarantees that any value read from memory will be the last value written
- Changes migration trust model
- How can we make sure that a person in the middle can’t drop or replay some of the pages?
- How can we make sure that the HV sends all the necessary pages?

- Migration Agent & Initial Migration Image

Open Questions

- Post-copy
- Parallelism for Migration Handler
- Generalized confidential migration

KVM FORUM