Attestation and Secret Injection
for Confidential VMs, Containers and Pods

Jim Cadden, James Bottomley
IBM Research
September 21st, 2021
AMD SEV Confidential Computing hardware provides methods for:

1. **Attestation/Measurement**: allows the external guest owner to validate the contents of the guest VM prior to exposing sensitive data.

2. **Secret Injection**: allows the external guest owner to inject secrets into the VM without the host/hypervisor being able to read it.
Prelaunch Attestation in SEV and SEV-ES

• The measurement validation and secret injection happens prior to the launch of the guest

• Secure hardware measures the initial memory (FLASH0 only) and exports it via a secure channel to the guest

• External guest agent validates the measurement against an expected value, return a secret “table” in reply
Secret Inject in SEV and SEV-ES

- Secret table is injected into EFI-provided memory where it can be read by the guest bootloader or OS
  - Secrets can be symmetric keys, API credentials, asymmetric keys
- Confidential software components must be gated by a secret
- Secrets ensures *ownership* of the guest validation, otherwise an attestation report can be falsified or ignored by a malicious host/hypervisor
Confidential Boot of an Encrypted Disk

- Guest owner encrypt their disk image root partition (LUKS + QCOW2)
  - Existing QEMU support for disk encryption breaks the SEV trust model
- At launch, decryption key is passed by the guest owner into guest firmware memory
- GRUB moved into the firmware so it can be attested
- GRUB retrieves key and decrypts root partition for boot (DMCrypt)
  - Functionality merged into upstream OVMF, Qemu, Linux (ongoing)
Confidential Boot of a Container / Pod

- Kata Containers specifies a kernel, initrd, and command line for each VM. These file are selected and managed by the untrusted host!

- Our Solution:
  - On launch, hypervisor computes the hashes of kernel, initrd & command-line and writes them into guest’s initial memory
  - Hashes are then contained within the attestation measurement validated by the guest owner
  - OVMF confirms the hashes computed by the hypervisor match those the files about to be run
Secret Injection for Kata Containers

- Secret table is moved from EFI memory to kernel reserved memory
- Kernel module, loaded during init, creates SecurityFS entries for each GUID in the secret table
- Kata attestation-agent reads secrets via `/sys/kernel/security/coco/sev_secret/`
  - Symlink points to a hardcoded GUID that contains the metadata of the table
- Secrets can take many forms:
  - Set(s) of keys for container image layer decryptions
  - Key to establish external credentials (e.g., API server)
  - Container *allowlists*
Container Attestation-Agent

- *Hardware-agnostic* attestation components for encrypted container images
- For each layer of an OCI-encrypted image the decryption library receives a key from a local attestation-agent
- Attestation agent pulls keys from secret table (sev:SecurityFS )

- Early & Active & development:
  - https://github.com/containers/attestation-agent/
Thank You!

• This concludes my presentation
1. Launch Container

2. Store key in HSM
   - HSM

3. Pull encryption tokens

4. Build launch data
   - Launch Data:
     - launch_blob.bin
     - godh.crt

5. Pull Guest Credentials
   - platform:
     - poh.crt
     - chain.crt
     - hw.info

6. Launch SEV VM

7. Read measurement.bin

8. Generate secret

9. Pull secret
   - GenCTC
   - packaged_secret.bin

10. Inject secret