Full disk encryption for Confidential Computing guests

FOSDEM 2026

Emanuele Giuseppe Esposito

Vitaly Kuznetsov

Confidential computing

Confidential VMs ares here to stay, you can easily get one from AWS/Google/Azure or deploy your own KVM + SEV-SNP/TDX

- TDX, SEV-SNP reliably protect the data in use.
- Protecting the storage in left to the guest operating system.

Confidential storage

Confidential storage must provide:

- Verity protection for readonly parts.
- Encryption + integrity protection for read-write parts.
- Attestability.
- Roll-back attack protection.

Verity protection

- Dm-verity is great!
  - integrity checking for block devices
- Rich support in systemd:
  - repart, veritysetup-generator, dissect,...
  - ***roothash=<hash>*** for full root protection in read-only or with ephemeral overlay (systemd.volatile=overlay)
  - ***usrhash=<hash>*** for /usr protection (aka hermetic /usr approach)
- Downside: verity target partition is RO!

Encryption

- LUKS is great! Partition is encrypted and RW!
- Each VM instance/volume needs to be individually encrypted in a safe environment:
    - Pre-encrypted by some ‘trusted’ infrastructure.
    - Self-encryption e.g. upon the first boot.

Encryption and systemd-repart

- Systemd-repart can create encrypted volumes and seal keys to the vTPM
  - Integrity protection for LUKS (experimental) is coming in v260
    - Full roll-back attack always remains possible.
  - A new feature to ensure that the created encrypted volume will not change before (first) usage.

Combining encryption with verity

Verity -> Encryption switch can give us ‘read-write experience’:

- Copy everything from verity-protected volume to the encrypted one.
- Use filesystem overlay (overlayfs) on the encrypted volume over verity-protected data.
- Use dm-clone and transfer verity-protected data to the encrypted volume.

Verity + encrypted overlay

- ***roothash=<hash>***
  - Root is found via roothash, verified and mounted
  - Currently, it can’t be combined with a persistent overlay with standard systemd tooling to give read-write experience.
- ***usrhash=<hash>:*** a read-write solution is coming
  - /usr is separate, dm-verity protected partition
  - Root is created by systemd-repart and encrypted
  - /usr is mounted ro (verity) and an overlay makes it rw

Dm-clone approach

- *Almost* no influence on the boot time (depends on the hydration parameters).
- Can simplify things in case of complex storage configurations, e.g. LVM.
- Potentially allows to minimize storage use by dropping dm-verity protected data after full convergence.
- No support in the standard tooling but a solution is being work on.

Encryption and attestation

- Systemd already measures a derivation of LUKS volume key to PCR15.
  - *… alongside randomly generated machine-id*
- For self-encryption, a proof that the volume was created in a safe environment (e.g. on the first boot) is also needed.
  - See the proposal!

EFI system partition

- Cannot be verity protected and/or encrypted.
- SecureBoot keys (+ Measured boot) need to be trusted for ESP artifacts:
  - UKI
  - Cmdline extensions
  - Systemd sysext/confext

Using distro-shipped UKIs

- Expected verity hash (***roothash=…/ usrhash=…)*** can be supplied with a cmdline extension.
- Encryption logic can be a systemd sysext/confext:
  - Distros using dracut may need [a new feature]

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/@redhat

facebook.com/RedHat

x.com/RedHat