MachineOS: a Trusted, SecureBoot Image-based Container OS

Ryan Harper, Cisco

FOSDEM2023 2023-02-04

MachineOS

Designed[1] for appliances in lights-out/hands-off environments.

- Utilizes UEFI SecureBoot platform and TPM 2.0 devices
- Guards a Secure Unique Device Identity (UUID) in TPM for identity and authenticity
- TPM secrets only available to kernel/userspace if chain-of-trust is verified running signed software.
- Supports unattended encrypted storage for at-rest protection of device data
- Continuous and Incrementation Updates

1. Securing TPM Secrets in the Datacenter LSS2021 P.Moore, J. Latten

Root of Trust - Secure Unique Device Identity

- X.509v3 certificate and an associated key-pair which are protected in hardware
- Certificate contains the product identifier and serial number and is rooted in Public Key Infrastructure.
- Key pair and a certificate are inserted into hardware during manufacturing.
- The certificate provides an immutable identity for the device that is used to verify that the device is a genuine product, and to ensure that the device is well-known to the customer's inventory system.

MachineOS - Product Runtime

```
+-----------------+           +-----------------+
| Platform Device |           |    mOS UKI      |
+-----------------+           +-----------------+
|                 | Verify    |                 |
| TPM: LOCKED     +---------->|  kernel+initrd  |
| Svc: inactive   |SecureBoot |  +cmdline       |
|                 |           |                 |
+-----------------+           +--------+--------+
         ^                             |
         |              Load EA Policy |
  Update |              Verify PCR7    |
  Reboot |              Verify Version |
         |                             V
+--------+--------+           +-----------------+
| Platform Device |           | Platform Device |
+-----------------+ Read Rot  +-----------------+
|                 |  Extend   |                 |
| TPM: LOCKED     |<----------+ TPM: unlocked   |
| Svc: Active     |  Decrypt  | Svc: inactive   |
|                 |  Verify   |                 |
+-----------------+           +-----------------+
```

MachineOS - Working with MachineOS

```
                                           +----------------------+
                                           |   Least Frequent     |
                                           +----------------------+
+--------------------+                               |
|Keys and Certs      |                               |
+--------------------+                               |
                                                     |
 Generate RoT Cert -> Generate Product Keys          |
                                                     |
+--------------------+                               |
|Platform            |                               |
+--------------------+                               |
                                                     |
 Configure Firmware -> Provision TPM                 |
                                                     |
+--------------------+                               |
|Product Image       |                               |
+--------------------+                               |
                                                     |
 Pull -> Verify -> Sign -> Generate                  |
                                                     |
+--------------------+                               |
| OCIs               |                               |
+--------------------+                               |
                                                     |
 Build -> Sign -> Publish                            |
                                                     V
                                           +----------------------+
                                           |   Most Frequent      |
                                           +----------------------+
```

MachineOS - OCI Build, Sign, Publish

```
+--------------+                       +------------+
|              |    stacker build      |            |
| image1 yaml  +----+----------------> |            |                      ZOT
|              |    |                  |   OCI-1    |                 +-------------+
|              |    |                  |            +---+             |             |
+--------------+    |                  |            |   |             |             |
+--------------+    |                  +------------+   |             |             |
|              |    |            +---->| signed sha |   |             |             |
| image2 yaml  +----+            |     +------------+   |  publish    |             |
|              |                 |                      +------------>|             |
|              |                 |     +------------+   |             |             |
+--------------+                 |     |            |   |             |             |
                                 |     |            |   |             |             |
+---------------+                |     |   OCI-2    +---+             |             |
|    TRUST      +----------------+     |            |                 |             |
|               |     sign       |     |            |                 +-------------+
|  user.key     |                |     +------------+
|  user.cert    |                +---->| signed sha |
|               |                      +------------+
+---------------+
```

MachineOS Lifecycle - Product Image Build

```
 pub/priv zothub.io            MyZotService.yaml             mOS Service Image
+--------------------------+  +--------------------------+  +----------------------------+
|                          |  |                          |  |                            |
|  +--------------------+  |  | - src: zot://mos:v1       |  | +------------------------+ |
|  |  mOS v1.0 UKI OCI  |  |  | - src: zot://zot:v1.4.3   |  | | OCI Layout             | |
|  |                    |  |  |                          |  | | +--------------------+ | |
|  |                    |  |  +------------+-------------+  | | | OCI:mos:v1         | | |
|  |                    |  |               |                | | +--------------------+ | |
|  |                    |  |               |                | | | user pubkey sig    | | |
|  +--------------------+  |               |                | | +--------------------+ | |
|  |  SHA256SUM  A      |  |          +----v-------+        | |                        | |
|  +--------------------+  |          |            |        | | +--------------------+ | |
|  |  mOS pubkey sig    |  |  Pull    | ImgBuilder | Build  | | | OCI:zot:v1.4.3     | | |
|  +--------------------+--+--------> |            +--------> | +--------------------+ | |
|                          |  Verify  |            | Sign   | | | user pubkey sig    | | |
|  +--------------------+--+--------> |            |        | | +--------------------+ | |
|  |  zot v1.4.3 OCI    |  |          |            |        | |                        | |
|  |                    |  |          +----^-------+        | |                        | |
|  |                    |  |               |                | +------------------------+ |
|  |                    |  |               |                |                            |
|  |                    |  |               |                | +----------------------+   |
|  +--------------------+  |       +-------+-------+        | | myzotservice.yaml    |   |
|  |  SHA256SUM  B      |  |       |    TRUST      |        | +----------------------+   |
|  +--------------------+  |       |               |        | | user pubkey sig      |   |
|  |  zot pubkey sig    |  |       |  user.key     |        | +----------------------+   |
|  +--------------------+  |       |  user.cert    |        |                            |
|                          |       |               |        |                            |
+--------------------------+       +---------------+        +----------------------------+
```

MachineOS Components – Runtime

Runtime Image

- Single UEFI SecureBoot Image (UKI-like)
  - Stubby
  - Linux kernel
  - Initrd with MachineOS tools
  - Embedded kernel command line
  - Signatures/Certs
- One or more signed OCI Container images

Runtime Tools

Tools for verification and execution of signed OCI

- mosctl
  - install mOS system
  - verify and start containers/services
  - update mOS system
- trust
  - Provision TPM with secrets
  - Access secrets, certs, key pairs
  - Load TPM EA Policies
  - Extend PCRs during early boot

MachineOS Components – Build Tools

Build Platform Tools

Build MachineOS OCIs and run clusters of mOS workloads

- stacker
  - build OCIs using signed SquashFS layers annotated with verity hashes
- machine
  - Run instances of mOS (baremetal, virtual)

Anatomy of UKI-like boot image

Consolidate the kernel, initramfs and signatures into single EFI application.

- Tightly couple kernel, initramfs and cmdline for security
- Fewer moving parts for updates
- Restricted kernel command line with some flexibility for certain parameters
- Signed with Product release key, verified with Product certificate which contains the device UUID that must match the UUID in RoT certificate.

```
.----------.
|          |
| stubby   | <--- Verifies cmdline from UKI or boot entry
|          |
|----------|
|          |
| kernel   | <---.
|          |     |
|----------|     |
|          |     |
| initrd   |     |
|          |     |--- stubby loads into memory
|----------|     |
|          |     |
| cmdline  |     |
|          | <---'
|----------|
|          |
| sigs     | <--- Read by shim and extended into PCR7
|          |      Written with sbsign
`----------'
```

Boot Process - Hardware -> Firmware -> Bootloader

- Firmware signatures checked by hardware platform
- UEFI SecureBoot
  - UEFI verifies bootloader has valid signature found in firmware key database
  - Once verified bootloader will execute with the path to the UKI provided as input
- Bootloader
  - Verifies signature of the UKI with UEFI and shim key databases
  - Extend PCR7 with the signing key/cert measurements
  - Validates cmdline (passed in boot entry or built-in to UKI)
  - Execute loaded kernel,initramfs with cmdline

Boot Process - Kernel -> Early Userspace Accessing TPM

- Normal kernel boot and transition into initramfs's /sbin/init
- Load the EA policies into TPM and attempt to read secrets
- TPM NVIndicies unlocked if PCR7 matches and Version NVIndex is correct for the loaded policy. On failure, system is halted.
- Load RoT keys, certs, and mOS LUKS passphrase into kernel keyring readable only by root.
- Extend PCR7 with well-known measurement sealing further access to protected TPM NVIndicies

Boot Process - Verifying Product and Starting Workload

Verifying Product Manifest

- Mount encrypted filesytem using LUKS key extracted from TPM
- Verify signature of Product manifest using Product certificate stored in LUKS encrypted filesystem
- Verify Product manifest certificate is signed by manifestCA included in initrd
- Verify signature of Product manifest using Product certificate
- Verify Product manifest certificate is signed by CA included in initrd

Activating Workload

- Use Product manifest to mount dmverity-protected OCIs and service containers
- Optionally pivot-root into a 'boot' OCI
- Start containers

Verifying OCIs with DM-Verity

- OCIs built by stacker using SquashFS layers are annotated with verity hashes.
- Mount OCI via cryptsetup/dm-verity using hashes stored in annotation
- As OCI data is accessed through dm-verity block device, the kernel will verify the integrity of the data against the loaded hash.
- Failed verification results in I/O errors

Later today Scott Moser - Secure Container Storage @ Containers DevRoom @ 16:05

MachineOS - Incremental and Continus Updates

Updates may upgrade or downgrade system by loading a new manifest specifying different OCI images.

mOS must perform the same verification of the update as is done during initial boot, namely verifying signatures on the manifest and on the OCI images included.

- Update configuration data with pointer to the specified version of image
- Restart Updated Containers
- If UKI or other boot/rootfs changes, reboot system.

mOS can be configured to sync from an OCI registry e.g. zot via `zot-sync` to keep the system patched and up-to-date.

Revocations via Policy Version update

mOS uses TPM Extended Authorizations (EA) policy to gate access to secrets and leverages a Policy Version NVIndex in addition to PCR7 to restrict access between Production and Management/Owner.

TPM access to LUKS and RoT keys involve two steps:

- Verify PCR7 matches expected values
- Verify NVIndex holding 'EA policy version' value is at the expected value.

Production/Runtime UKIs do not have access to the TPM owner/admin password which is stored in the TPM itself during provisioning[1]. A separate UKI signing key and EA policy is used in a restricted management EFI application which can be used to increment the Policy Version NVIndex and prevent previous UKIs from being able to run and unlock access to the TPM.

Future work evaluating use of TPM counting indicies as proposed in the UAPI SecureBoot an alternative to using a secondary key/mgmt EFI app which must also be kept secret.

1. Securing TPM Secrets in the Datacenter LSS2021 P.Moore, J. Latten

Tools for building mOS and Container images

- project-stacker[1][2]
    - Create OCIs with SquashFS layers
    - Annotated OCI with dm-verity hashes in annotations
- project-zot[2][3]
    - OCI Native Image Repository
    - CNCF Sandbox Project
    - Unprivileged single binary
    - Multi-arch, Multi-os
    - scanning, authentication, authorization, dedup included

1. https://github.com/project-stacker/stacker
2. https://stackerbuild.io
3. https://github.com/project-zot/zot
4. https://zotregistry.io

Questions

Connect with Project Machine @ Project Machine Discussions

Thank You

- Serge Hallyn, Cisco
- Joy Latten, Cisco
- Scott Moser, Cisco Secure Container Storage
- Paul Moore, Microsoft

Project Machine