Image-Based Linux and Secure Measured Boot Devroom Intro

or

UKI? DDI?? Oh my!!!

FOSDEM 2023
Luca Boccassi, Linux Systems Group, Microsoft

Welcome to the devroom!

- Huge thanks to organizers and contributors
  - Thilo Fromm
  - Zbigniew Jędrzejewski-Szmek
  - Mathieu Tortuyaux
  - Kai Lüke
  - Morten Linderud
  - …and probably more
- Devroom logistics
  - 10m break at 12:10, finish at 14:20
  - Recording/live streaming

I’ve seen this before…

- Embedded folks have been doing image-based Linux for decades
- Our focus is on security, measurability, attestation rather than size/hardware
- First-class support for at least one of UEFI Secure Boot or TPM-based measurements, most often both
- Extend chain of trust (firmware -> kernel) to userspace (initrd + root FS)
  - Sign initrd
  - Protect root FS
  - Hermetic /usr (merged-usr)
- UAPI Group
  - https://uapi-group.org/
  - https://github.com/uapi-group

But wait, there’s more!

- At least three different philosophies for immutable image-based OS
- GPT/raw images
  - build images remotely
  - dm-verity, read-only volumes installed with A/B schemes
- (RPM) OSTree
  - build (packages or) OSTree snapshots remotely
  - apply changes/switch snapshots locally, read-only/ephemeral at runtime
- BTRFS
  - build packages remotely
  - apply changes/switch BTRFS snapshots locally, read-only/ephemeral at runtime
- Different implementations, but shared goals, tools, specs

UKI: Unified Kernel Image

- UEFI stub + Kernel + initrd [+ cmdline [+ osrelease [+dtb …]]]
- Built via objcopy or ukify
- Single PE binary
- Signed for Secure Boot
- Installed in ESP/XBOOTLDR
- Auto-discovered by bootloaders implementing BLS
    - https://uapi-group.org/specifications/specs/boot_loader_specification/
- Predictably measured into TPM (PCR11)
- Future work: support multiple command line entries
- https://uapi-group.org/specifications/specs/unified_kernel_image/

DDI: Discoverable Disk Image

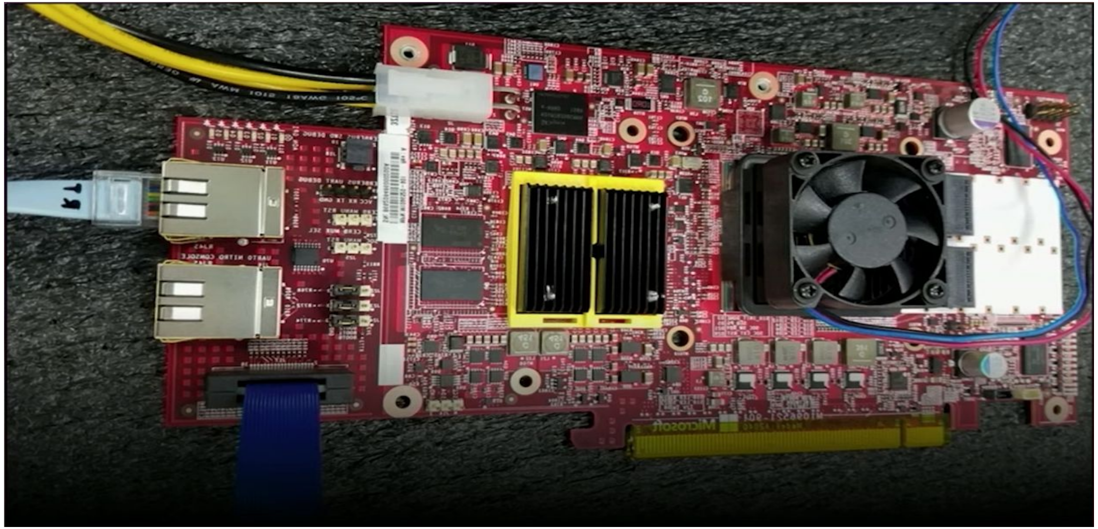
- Raw disk image, self-described by GPT partition table following DPS
- Partitions are tagged with well-known GUID depending on purpose/mount
  - https://uapi-group.org/specifications/specs/discoverable_partitions_specification/
- Natively supports signed dm-verity protection for root/usr partitions
- Upcoming feature: user can impose requirements, eg: DDI must have verity
- Same DDI can be used by different tools without any changes, e.g. for root:
  - If it's on the disk where the ESP is located at boot, systemd will use it as the OS root FS
  - If it's passed to nspawn, it will be used as the container's root FS
  - If it's passed to portabled, it will be used as the portable service's filesystem
  - If it's passed to systemd-sysext, it will be used to extend the root FS
- https://uapi-group.org/specifications/specs/discoverable_disk_image/

sysext: system extension DDI

- An interesting form of DDI: sysext, can be used to securely extend a root FS
- Contains /usr and (optionally) /opt hierarchies - single tree for each vendor
- Identified by /usr/lib/extension-release.d/extension-release.$image
  - https://www.freedesktop.org/software/systemd/man/os-release.html#/usr/lib/extension-release.d/extension-release.IMAGE
- Root FS DDI + bunch of sysext DDIs = read-only OverlayFS on /
- As a DDI, sysext can be protected by signed dm-verity
- As a DDI, sysext can be passed to different tools
  - If it’s on the disk where the ESP is located at boot, systemd will use it to extend initrd
  - If it’s in /var or /etc, systemd will use it to extend root FS
  - If it’s passed to portabled, it will be used to extend the portable service’s filesystem
- https://uapi-group.org/specifications/specs/sysext/

This stuff is real, I swear

- Real-world use case: Linux hardened OS for the ARM SoC in the Azure fleet
- Provides dedicated offloading and acceleration for Azure hosts
- Extensively uses DDIs (and soon UKIs if all goes well)
- https://www.youtube.com/watch?v=PO5ijv6WDv0&t=608s



Thanks!

- Come talk to us, we don’t bite (unless we are hungry)
  - https://uapi-group.org/
  - https://github.com/uapi-group
  - Join us and embrace a more secure way of doing Linux
  - Help us extend the specifications
  - Ultimate goal is to get a whole class of security bugs extinguished
- Questions?