GRUB2 security features overview

GRUB2 and 3mdeb "minisummit"

Piotr Król

3MDEB

Agenda

- Authentication and authorisation in GRUB
- Using digital signatures in GRUB2
- UEFI secure boot and shim support
- Measuring boot components
- LUKS/LUKS2
- Reproducible builds
- GRUB2 integration for validation frameworks
- HTTPS booting

Authentication and authorisation in GRUB

- works as documented through
- probably most useful use case from embedded perspective would be to block CLI and editing

- minor issue with this features is that you don't really know what is going on since after failure there are no messages

Using digital signatures in GRUB2

- First thing that is interesting is support for DSA keys, even in light of deprecating those by OpenSSH
  - this can be controversial, so crypto experts should express opinion
- Potentially it could be useful to verify signatures using TPM
- Build process also should support signing
  - there is no way to add during build in GRUB2
- Definitely feature is not easy to use, some people simplify process by adding scripts to system
  - https://github.com/Bandie/grub2-signing-extension
- There are also some problems with applying description form documentation about and option:
  - https://unix.stackexchange.com/q/531992/44718
- Question is who use this feature and how?

Using digital signatures in GRUB2

- continuation of vboot signature verification process beyond bootloader for coreboot supported systems
- there is also other issues in whole feature and its use, which is GPLv3
  - this is not legal meeting
  - keys have to be revealed to anyone who own the device
  - how we can reliably proof device ownership?
- We believe that in long run very important would be process of secure and privacy-aware device re-ownership
  - maybe in long run GRUB2 should simplify implementation around signing process to simplify that use case

UEFI secure boot and shim support

- There is seem to be very little information about that feature in GRUB2 manual
  - at least some references would be useful
- Does it make sense to support the same mechanics for coreboot?

Measuring boot components

- Only supported on UEFI platforms
- What should be first step to support measured boot on non-UEFI platforms?
    - INT 1Ah - makes sense only if first stage or coreboot expose that interface
    - TPM SPI driver - there is only one SPI driver so far (?) for RK3288
    - TPM LPC driver - there are some security concerns
    - TPMGenie resistant communication would be needed

LUKS/LUKS2

- There seem to be no official guide of using GRUB2 with cryptsetup
- Most wanted: GRUB2+LUKS2+TPM2 (what a coincidence :)
- This is very TrustedGRUB2 related thing
  - most materials about LUKS+TPM direct to TrustedGRUB
- This support is still needed in GRUB2

TPM support

- legacy systems using INT 1Ah while in MBR
- preserving INT 1Ah interrupt handlers
- exposing INT 1Ah for further components e.g. FreeBSD?

Reproducible builds

- Are reproducible builds enabled by default in GRUB2?

GRUB2 integration for validation frameworks

- CHIPSEC (Python)
- BITS (Python)
- FWTS (C)
- UEFI testing framework (MicroPython)

HTTPS booting

- Are there any plans for that?
- Is that correct to assume that GRUB2 leverage UEFI drivers in network boot?
- UEFI has that support, but probably only recent releases
- TBH TFTP is way outdated for any reasonable use

Summary

- In general adoption of security features is slow because it is hard to set it up
- It is good that some distros simplify that and during installation handle e.g. encryption setup
- We think that security should be available to all users, not only those technically savvy
- Definitely it would be important to leverage public founding and events like hackathons and GSoC to improve GRUB2 security features adoption
- Wherever it is aligned with 3mdeb goals we would like to help in improving that state