Towards measured boot out of the box

Matthew Garrett
@mjg59  |  mjg59@coreos.com  |  coreos.com
Security of the boot chain is vital
UEFI Secure Boot
Various embedded solutions
Rely on security of firmware
No way to prove verification happened
Why does this matter?
Compromised servers
Modified laptops
Can’t protect against hardware attacks
...but we can cover most others
Trusted Platform Module
Small chip
Platform Configuration Registers
Measurement
\[ PCR_{\text{new}} = \text{hash}(PCR_{\text{old}} | \text{hash}(\text{data})) \]
Associated log
Trusted GRUB
(old and busted)
Rohde & Schwarz
(no UEFI support, not TPM2 support)
https://github.com/coreos/grub
What do we measure?
Traditional approach
Most components in separate PCRs
Need to re-use PCRs
Order of loading matters
Unimportant configuration changes alter values
Suboptimal
Use the logfile
Replay log to ensure it’s valid
Look at individual log entries
Two choices
Log entry contains description of binary and hash of binary
Log entry contains text and hash of text
Policy describes each binary
Policy describes regular expressions
Where does the policy come from?
CoreOS builds policy automatically on OS release
Problems:
Initramfs varies across systems
Reproducible initramfs builds
Generic initramfs
Where do we store boot data?
Use UEFI variables
Use TPM
Things to use the TPM for:
TPMTOTP
Disk encryption keys
SSH keys
Unseal/reseal
Doesn’t TXT make all of this easier?
(ha ha ha)
No secure boot support
Incompatible with runtime UEFI
Summary:
Ship bootloader support
Ship known-good measurements
Integration with firmware updates
Deterministic initramfs generation