How to create a trust anchor with coreboot.

Trusted Computing vs Authenticated Code Modules

Philipp Deppenwiese

About myself

- Member of a hackerspace in germany.
- 10 years of experience in it-security.
- Did a lot work on trusted computing and system security at my last job at Rohde and Schwarz Cybersecurity.
- I am a Gentoo user.
- Now I am a web developer and system administrator. 🦄

Basics

Important acronyms

TPM - Trusted Platform Module

TCB - Trusted Computing Base

PCR - Platform Configuration Register

ACM - Authenticated Code Modules

PKI - Public Key Infrastructure

TEE - Trusted Execution Environment

TPM

- Trusted Platform Modules are smartcards with extra feature set.
- Version 1.2 and 2.0 are out.
- www.trustedcomputinggroup.org does the specification and compliance.
- The authorization is done via ownership model. User can own the TPM.
- A TPM is always passive and not active !

TPM 1.2

- Created for Digital Rights Management but never used for it.
- Huge portests in the internet done by the FSF.
- TCG stepped back and modified the specification in order to provide an ownership model, DAA and revokable Endorsement Key in order to stop identification and provide full control.
- Algorithm sizes are limited RSA-2048 and SHA-1.
- There is one open source software stack.

TPM 1.2

TPM 2.0

- Mainly build for Microsoft! Compliance testsuite and everything else was designed for Windows usage only.
- Specification was removed shortly after it appeared. You can't find it on the internet.
- Supports modern cryptographic algorithms.

TPM 2.0

- Two software stacks. IBM and Intel.
- TPM architecture/hierachy got much more complex.
- Protected against bus attacks by having DH key exchange to establish a secure connection.

Authenticated Code Modules

- The idea of signing blobs.
- Sometimes used with manifests like in the android application world.
- Definition mainly used and introduced by Intel. But the technology exists since ages.
- Always used to establish a Secure Boot / Verified Boot.
- User can't claim ownership in the most cases. It depends on the implementation.

Concept

1. Trust Anchor ⚓

- Minimal trusted computing base.
- Protected against hardware attacks.
- Only one trust anchor if possible !
- Cryptographic functionality must be given.
- Ownership must be enforced.

2. Chain of Trust ⛓️

- Starts always with the Trust Anchor.
- Each chain element must be a minimal trusted computing base.
- Each chain element must be checked by the previous chain with a cryptographic mechanism before it gets executed.
- The chain of trust must be under control of the owner.

3. The last chain element 💻

- Must always ensure the protection of the executed code.
- Normally used in conjunction with full disk encryption.
- Is the step into the usability and can offer some sort of authentication for the end user.

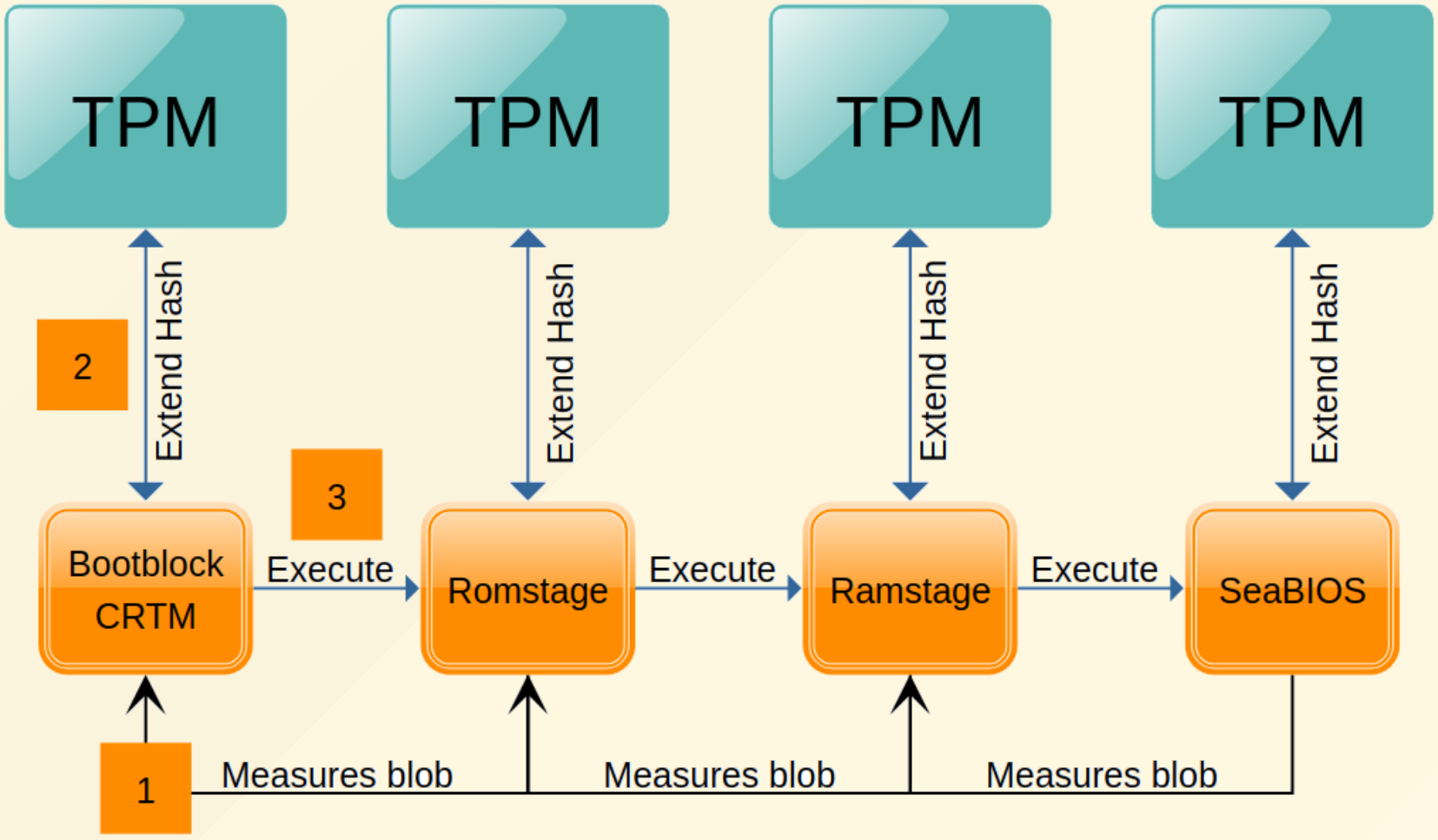
Implementation

Trusted Boot vs Secure Boot

The three principles of cryptography are integrity, authenticity and secrecy.

	Trusted Boot	Secure Boot
Integrity	yes (Sealing)	yes
Authenticity	yes (Sealing)	yes
Secrecy	yes (Sealing)	no
Freedom	yes	partial
Open Specification	yes	partial

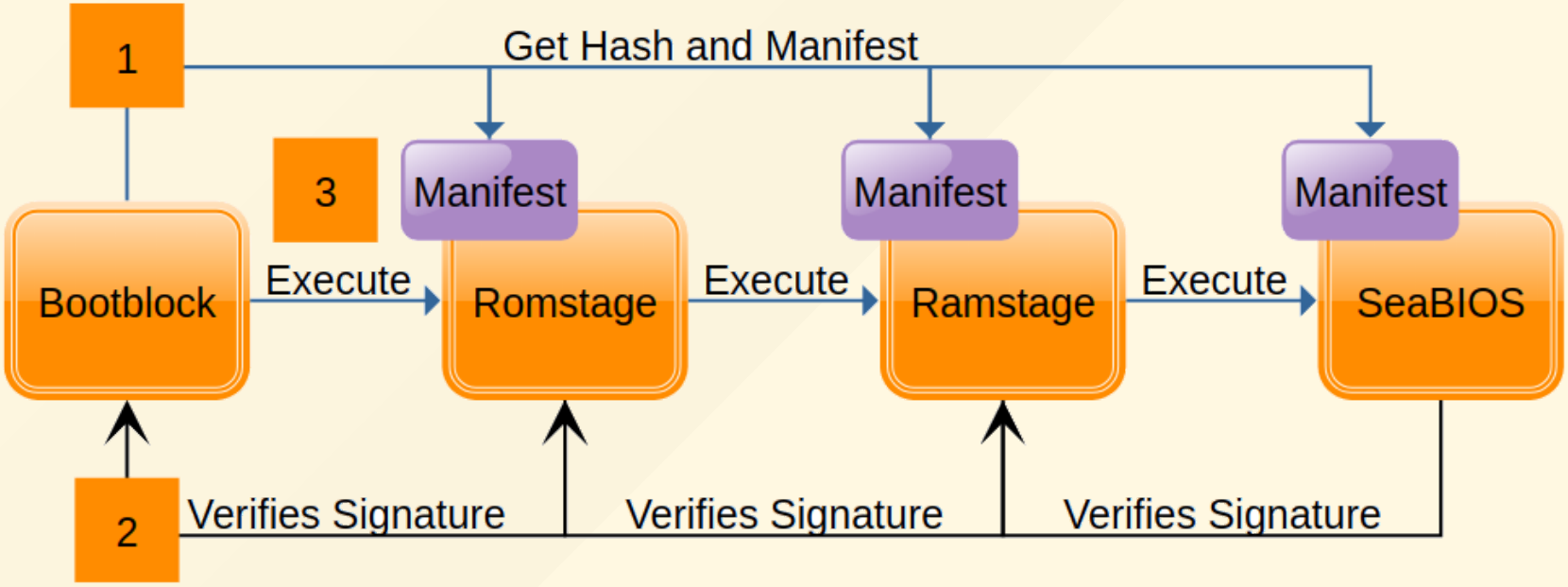
Trusted Boot



Trusted Boot

- Every hash is extended into a PCR. If there is already a hash do XOR operation.
- Measurements are done before code execution of the next chain element.
- Start with the Core Root of Trust for Measurement in the BIOS.
- Chain can be a SRTM or a DRTM. DRTM is a combination of ACM and Trusted Boot.
- Chain doesn't enforce anything.

Secure Boot


1
Get Hash and Manifest
3
Manifest
Manifest
Manifest
Bootblock
Execute
Romstage
Execute
Ramstage
Execute
SeaBIOS
2
Verifies Signature
Verifies Signature
Verifies Signature

Secure Boot

- Hash a blob. Sign the hash. Verify it.
- Verification is done before next stage is executed.
- Chain enforces ownership and error handling if a step can't be verified.
- Based on a complex PKI inside the firmware instead of a trust anchor.
- No hardware trust anchor except AMD PSP and Intel Bootguard.

Trusted Computing features.

Sealing/Binding

- Based on keys staying inside the TPM, like a smartcard.
- Padding scheme used is RSA-OAEP which is really important if it comes to sealing.
- **Binding:**
  -> Basically means using a TPM key for encrypting a blob.
- **Sealing:**
  -> Same as binding except adding the PCR into the cryptographic operation.

PCR and TCPA ACPI Log

- PCR are like slots for hashes.
- PCR can't be set to zero during runtime.
- TCPA log is used to document how the PCR values are calculated.
- Do **cat /sys/class/tpm/tpm0/pcrs** in order to get the PCR.
- Typically it can be found under **/sys/kernel/security/tpm0/ascii_bios_measurements**

Pratical application

- Create a trusted boot with sealing.
- Combine sealing/binding with disk encryption via hybrid encryption.
- OS can be secured with FDE and trusted boot.
- The last chain step will switch over to the big *TCB*.

Remote attestation

- Two methods can be used:
  -> Based on a third party model.
  -> Direct Anonymous Authentication.

0. EK certificate needs to be transferred in a trusted environment.
1. XOR all PCR to one hash with the TPM quote command.

Remote attestation

2. Sign the hash by an special AIK key which is signed by the EK too.
3. Send the quote data to an authetication server.
4. One unique hash in order to attest platform integrity. Authentication is done via unique EK certificate of the TPM itself.

Pratical application

- Allows us to prove that the platform has a specific state.
- If platform is safe then we can push credentials to it.
- There are different options to integrate a remote attestation:
  -> Using TLS by modifying the protocol.
  -> Using strongswan which is already integrated (TNC).

NVRAM, Monotonic Counter, TRNG

- Offers NVRAM which can be protected by the owner credential.
- Monotonic counter can be used againt replay attacks.
- True Random Number Generator.

coreboot and trusted computing

Basic Features

- coreboot offers support for TPM 1.2 and 2.0 .
- Different interfaces can be used: LPC, SPI and I2C.
- Easy integration through devicetree and kconfig.
- Startup and basic functions can be found.
- Not capable of having a trusted boot starting with the CRTM.
- No TCPA ACPI log filled by coreboot itself.

Two software stacks

- Google has it own software stack.
- Second software stack exists for basic usage.
- All software stacks are using the same drivers.

SeaBIOS to the rescue

- Offers support for TPM 1.2 and 2.0
- Includes a TPM menu for easy management via physical presence interface.
- TCGBIOS interface for bootloader usage.
- TCPA log is filled and done correctly.
- Measurements are done properly for everything which is executed by SeaBIOS.
- Since version 1.10.0 the TPM support is broken..

Flash protections needed

- Securing the trusted computing base. Includes coreboot + SeaBIOS.
- Protected Regions or SPI hw protections should be used.
- Not secure againt hardware attacks! Only if Bootguard or the PSP is used.
- If Bootguard is used I recommend the measured boot mode.

How to build a trusted boot

- Compile coreboot with TPM support and SeaBIOS.
- Claim the ownership on the platform.
- **Linux:** Install TrustedGrub2 from Rhode and Schwartz Cybersecurity https://github.com/Rohde-Schwarz-Cybersecurity/TrustedGRUB2
- **Windows:** Use Bitlocker which is already integrated in Windows.

Is a trusted boot really safe ?

- Evil Maid attacks are possible as long user interaction in the boot process is required.
- Can be migitated by STARK or MARK protocol.
- These attacks are only possible due to freedom of the boot process itself.
- If no user interaction is required the Evil Maid attack does not work.
- Hardware attacks are always possible.
- Using Intel Bootguard with measured mode and TPM 2.0 should make it really safe.

What's about updates in a trusted boot environment ?

- Normally you need to reboot. Bypass the sealing with a binding key..
- PCR pre-calculation is a solution.
- Look into the TCPA log or the firmware source code.
- There is no tool out there but it can simply be written ;) .

Is there an Open Source TPM ?

- Yes, buy a ARM Cortex-M4 board.
- Checkout https://chromium.googlesource.com/chromiumos/third_party/tpm2/ https://chromium.googlesource.com/chromiumos/third_party/cryptoc/ https://chromium.googlesource.com/chromiumos/platform/ec/
- Do some refactoring on the code.

Is there an Open Source TPM ?

- Compile and flash it.
- Thanks to Google for doing a great job open sourcing hardware as well.
- Have fun.

Conclusion

- Trusted Computing isn't so bad.
- Some nices features which can be used in order to attest the platform state.
- VBOOT2 is a combination of trusted and secure boot.
- Easy to use except the PCR pre-calculation.
- Open Source TPM is available soon.

Links and Sources

- https://sourceforge.net/projects/ibmtpm20tss/
- https://github.com/01org/TPM2.0-TSS
- http://trousers.sourceforge.net
- https://github.com/Rohde-Schwarz-Cybersecurity/TrustedGRUB2