TUX : Trust Update on Linux Kernel

Suhho Lee

Mobile OS Lab, Dankook university

suhho1993@gmail.com

--

Hyunik Kim, and Seehwan Yoo

{eternity13, seehwan.yoo}@dankook.ac.kr

Index

- Intro
- Background
- Threat Model
- Goals
- TUX : Trusted Update on Linux Kernel
- Experiments
- Discussion
- Conclusion

Intro

Security vulnerabilities are prevalent.

- Tons of CVEs
- Meltdown & Spectre

Lots of security updates!

Less attention was given to the changing integrity after the updates are conducted.

Intro

Maintain Integrity

Verifying the system by checking whether there are any malicious modification.

VS

Security Update

countermeasure vulnerabilities by modifying the system.

Intro

Can modern solutions distinguish between updates and malicious modifications?

Is the integrity being managed according to the updates?

Intel Trusted Execution Technology(TXT)

Intel's Measured / Verified boot technology

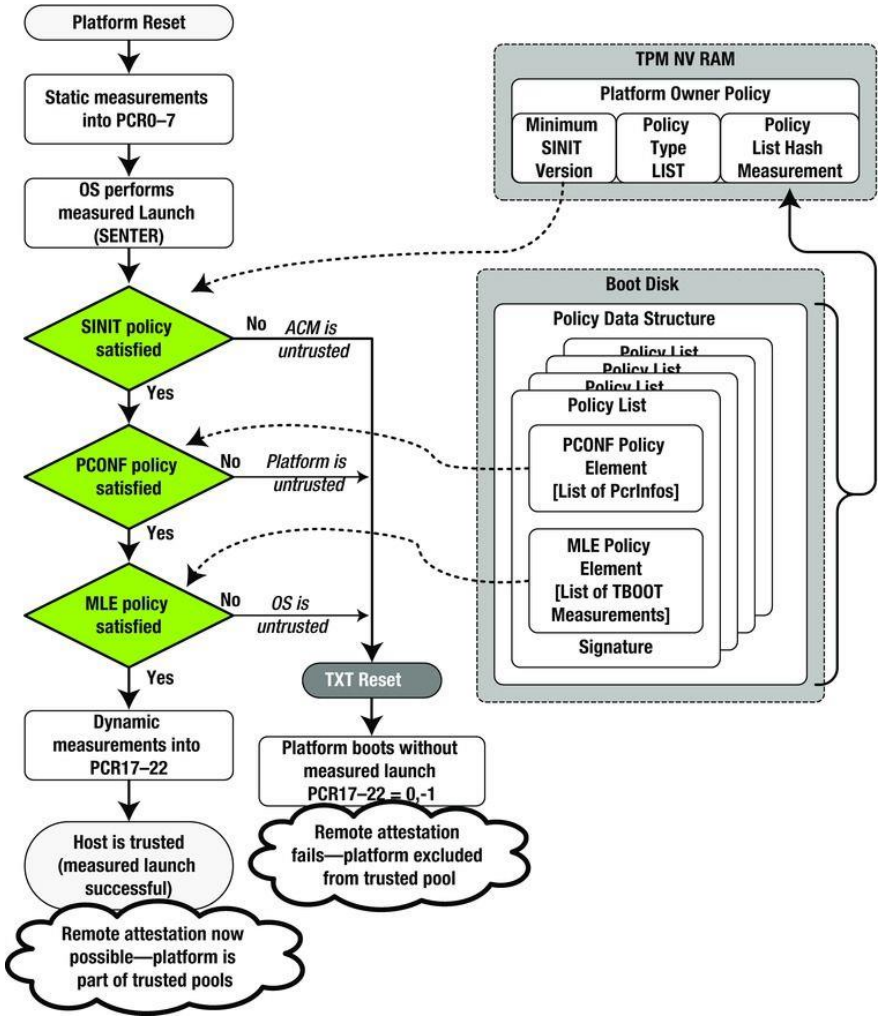
tboot

- Grub bootloader module
- Measure and verify integrity using DRTM

Launch control policy (LCP)

- Known good integrity values for certain booting stages
- Used to verify the integrity of the booting
- Also needs to be updated when the system is updated

Does not measure/verify Grub environment (e.g., Grub commands)



Verified boot using TXT

Open Cloud Integrity Technology (CIT)

Intel's remote attestation solution

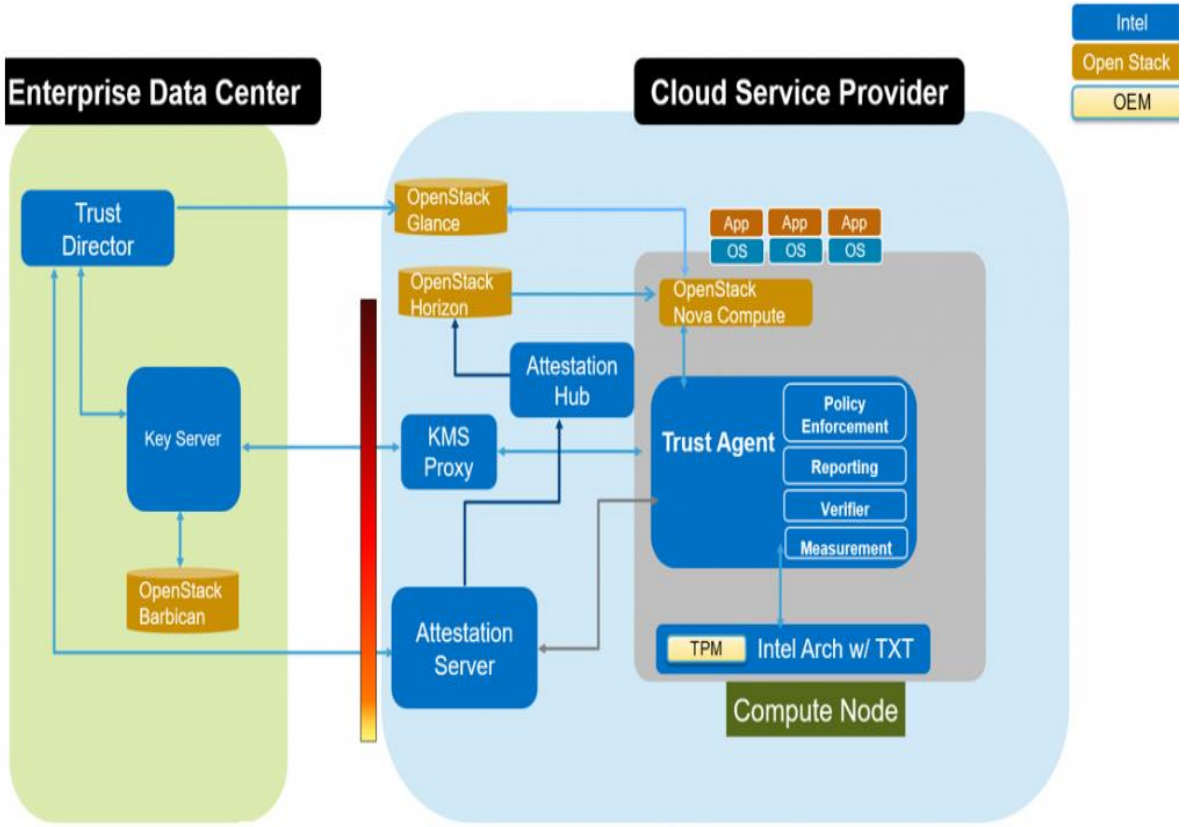
Validate TPM measurements from the remote server

Known-good values

- Stored in the Open CIT server
- Imported from the local systems

Local updates are not transparent

- Open CIT cannot monitor updates conducted from the local systems
- If local system is updated, remote attestation fails



Open CIT Architecture

UEFI secure boot

UEFI BIOS's Verified boot component

- Verify integrity using the key stored in the firmware DB
- Booting components must be digitally signed

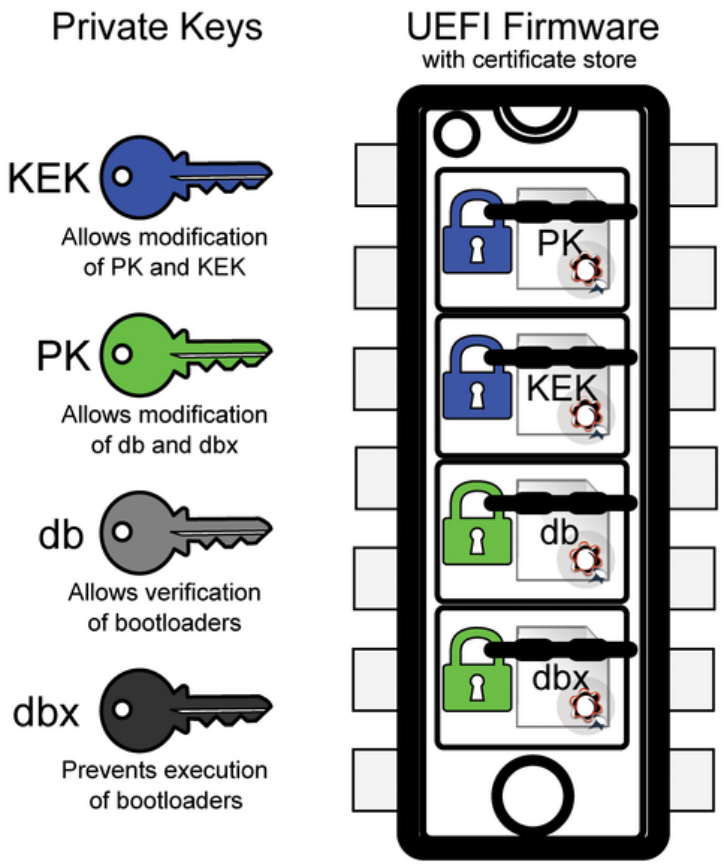
Updating is easy for Secure boot

- Sign updated binary and deploy
- Receive update and install

Not suitable for Linux environment

- Does not approve Grub bootloader
- Cannot verify Grub commands

No TPM measurements

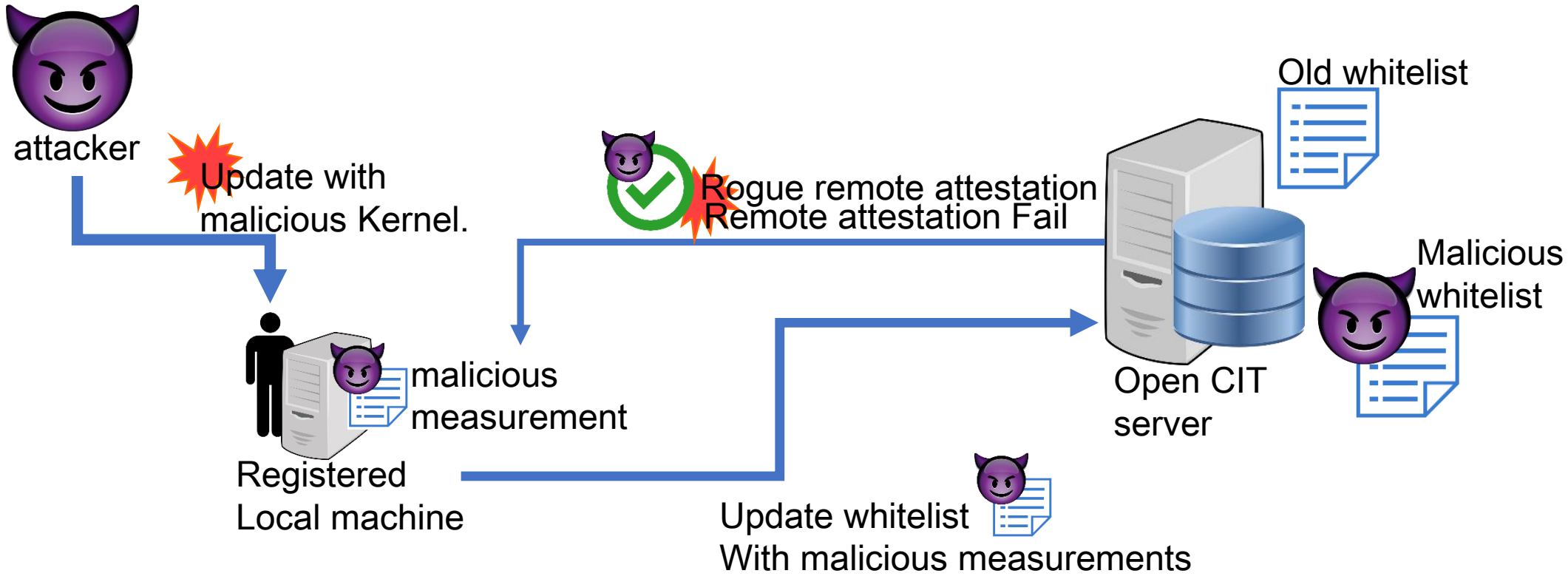


Keys used in UEFI secure boot

Threats!

Subverting Open CIT

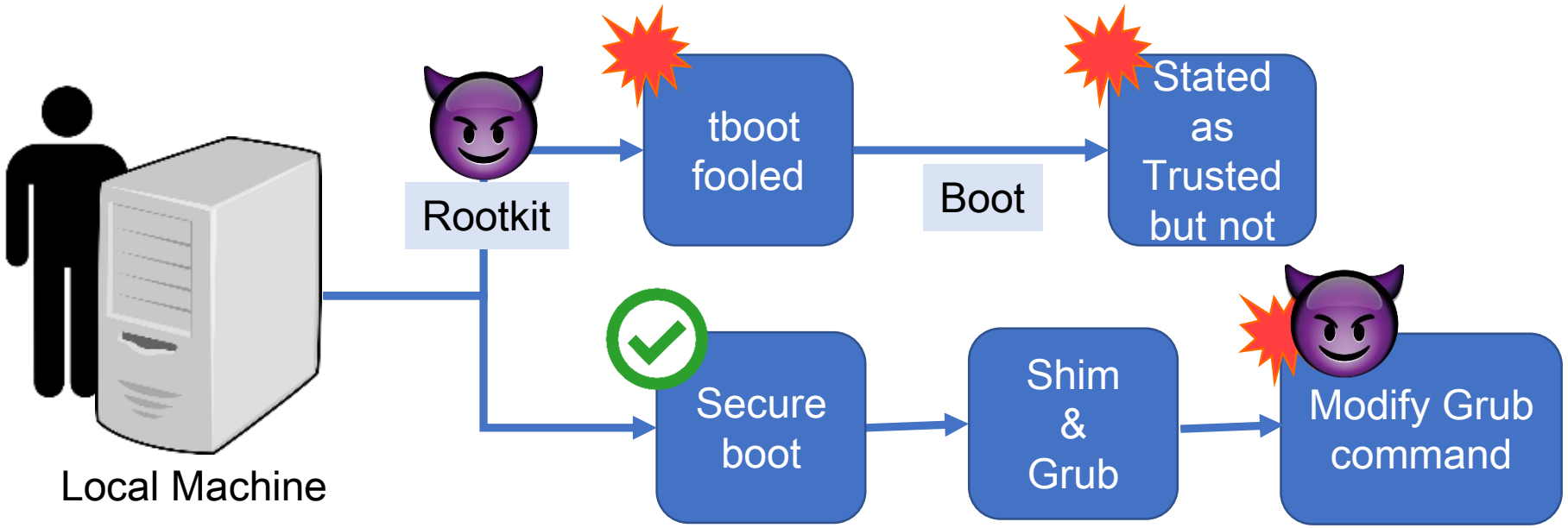
- Occurs because the local updates are not transparent to the Open CIT
- Assumption: Attacker can update OS and perform measured boot



Threats!

Circumventing verified boot

- Rootkit
  - Breaking Hardware-Enforced Security with Hypervisors (Black hat USA 2016)
- Attacker can modify Grub commands even though the Secure boot is on
- TXT and Secureboot lack of Grub command verification



Goals!

To maintain integrity properly...

1. Remote attestation must manage local updates transparently.
2. Maintain whitelist according to conducted updates and perform remote attestation using the up-to-date whites list.
3. Perform thorough measured / verified booting including Grub.

TUX!

TUX Code can be found at…
https://github.com/suhho1993/TUX.git

Trusted Platform Module (TPM)

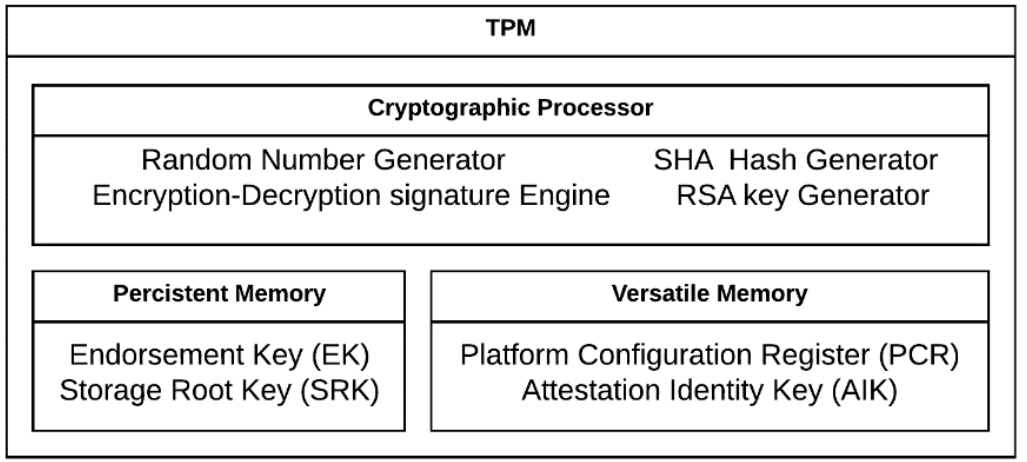
Provide TEE for integrity measurement

Tamper-proof device

Practically used in measured boot

Platform Configuration Registers (PCRs)

- TPM measures integrity using extend operation.
  - $PCR_{new} = Hash(\ PCR_{old} \oplus Hash(Data))$
  - Form Chain-of-Trust to verify the system configuration
  - Measure entire booting process with extend operation



TPM Architecture

TPM

Shim and Grub

Shim

- 1stage bootloader to support UEFI secure boot
- Can be verified by the UEFI secure boot
  - Signed with MS's firmware key
- Shim verifies and execute Grub
- Verification using firmware keys
  - Shim_lock verification

Grub

- CoreOS and other GRUBs support measured boot using TPM v2.0

Assumptions

TUX server is the maintainer/administrator of the updates.

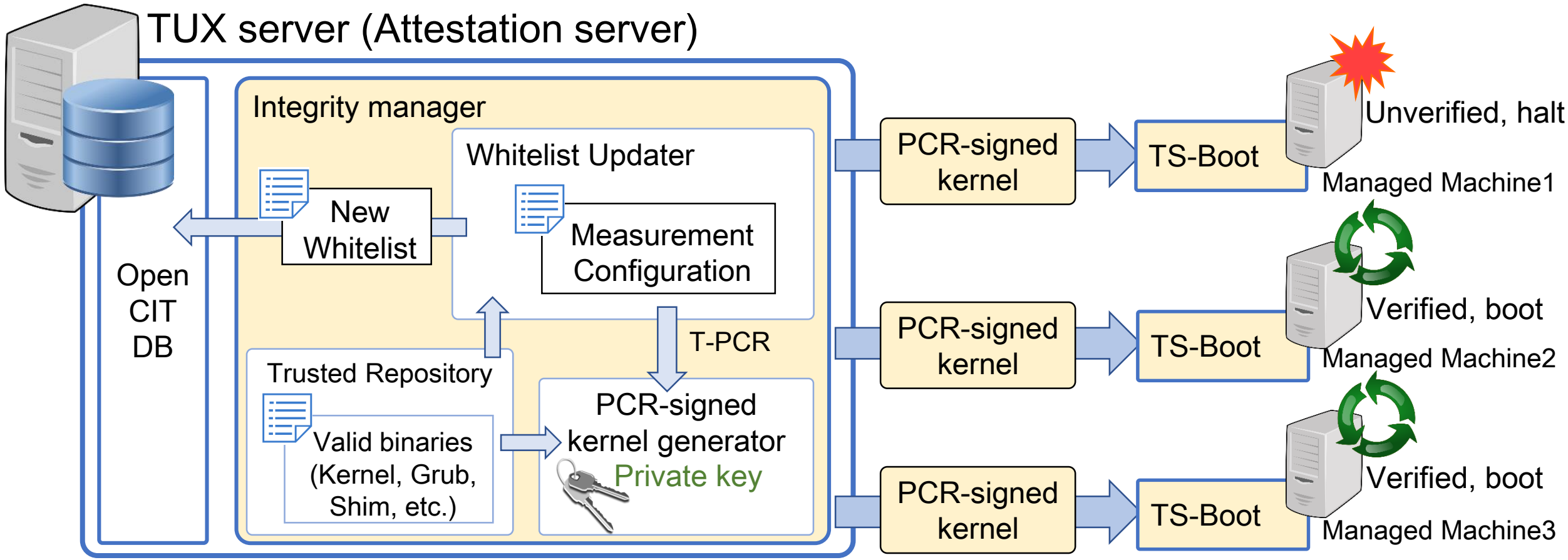
TUX only verifies integrity of the Linux Booting process.

TUX server is trusted and safe.

TUX owner holds manifest of specific booting process of each managed machine.

All managed machines hold TUX owner's public key.

TUX Architecture



Integrity Manager

Integrity management / kernel update component

- Located at the Open CIT server
- Consists of Trusted repository, Whitelist updater, PCR-signed Kernel generator

Trusted repository

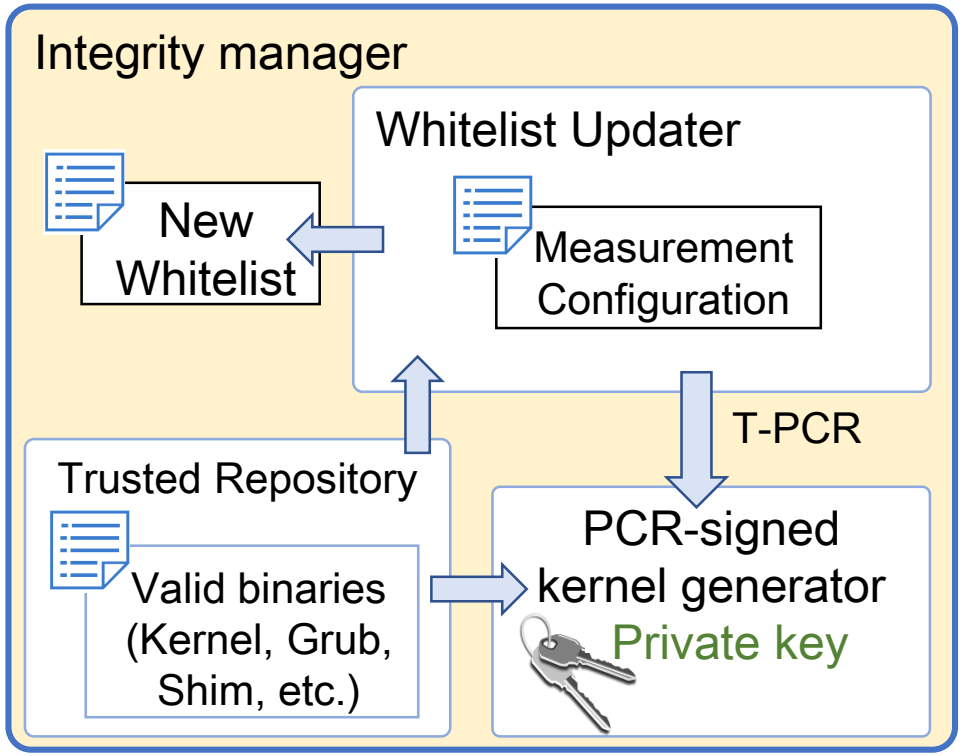
- Update repository
- Stores kernel binaries and manifests for the update
- Provide binaries to generate new whitelist.

Whitelist updater

- Calculate new integrity value and update the whitelist
- Calculate **t-PCR**

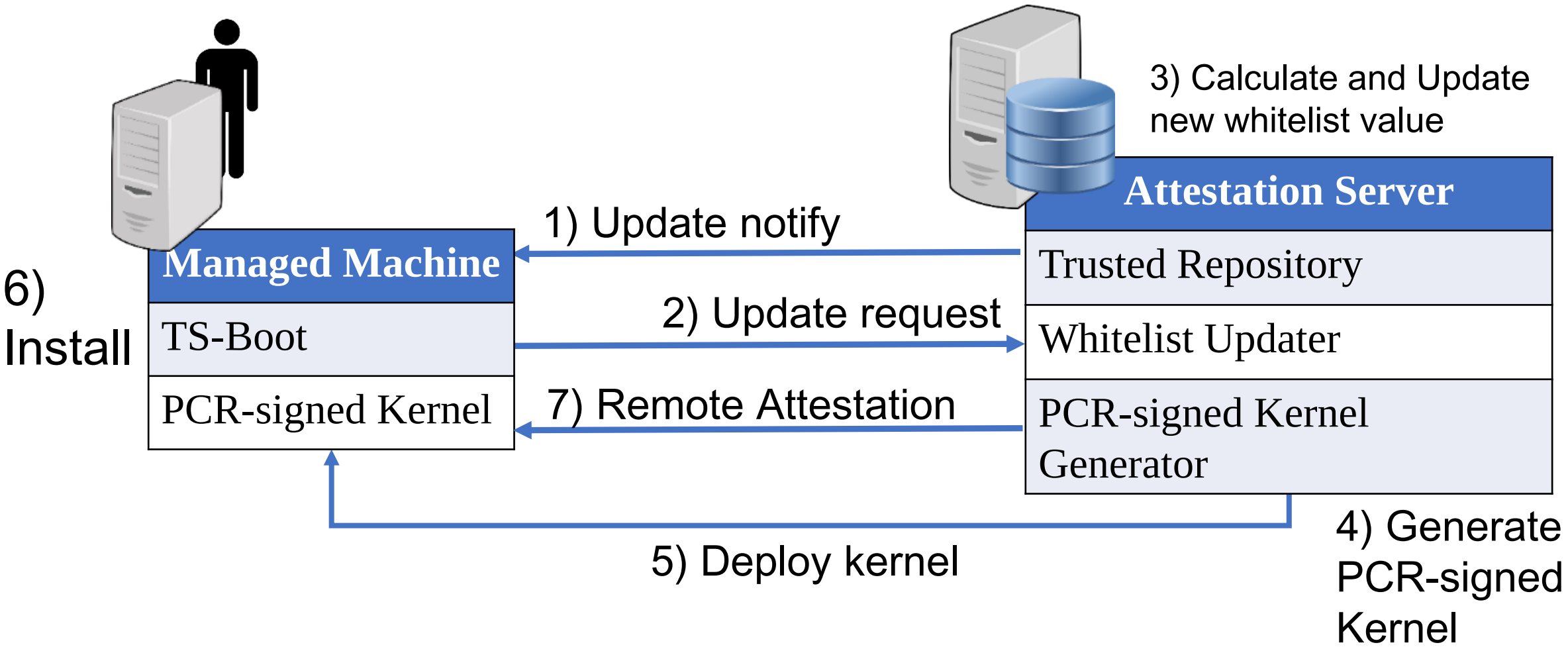
PCR-signed kernel generator

- Generate PCR-signed kernel



Integrity manager

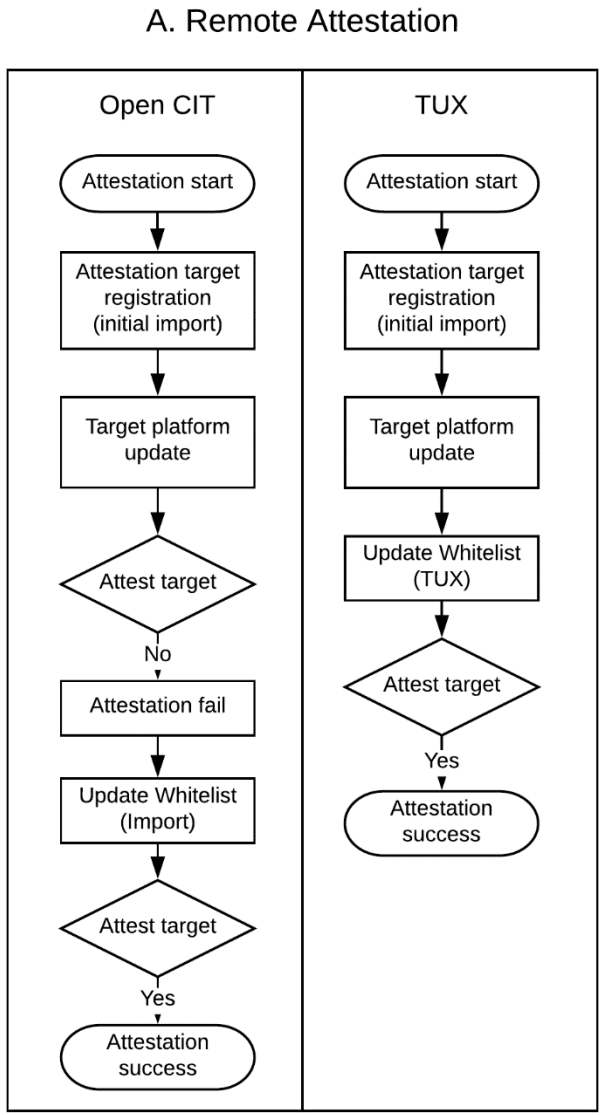
Kernel update using TUX



Remote attestation with TUX

With Integrity Manger, TUX...

- Manage local updates transparently (Goal 1)
- Prevent remote attestation failure after updates
- Perform remote attestation with updated whitelist (Goal 2)



PCR-Signed Kernel

t-PCR

- Calculated known-good value by extending entire booting process

PCR-signed Kernel

- Use t-PCR value to sign the kernel binary instead of digested hash
- Signed with TUX owner's private key
- Used to verify integrity during the booting

PCR-signed kernel

Trusted Secure boot (TS-Boot)

Combination of UEFI secure boot, Shim, and CoreOS Grub

- Perform robust measured/verified boot

Linux friendly

Binary verification using the firmware key

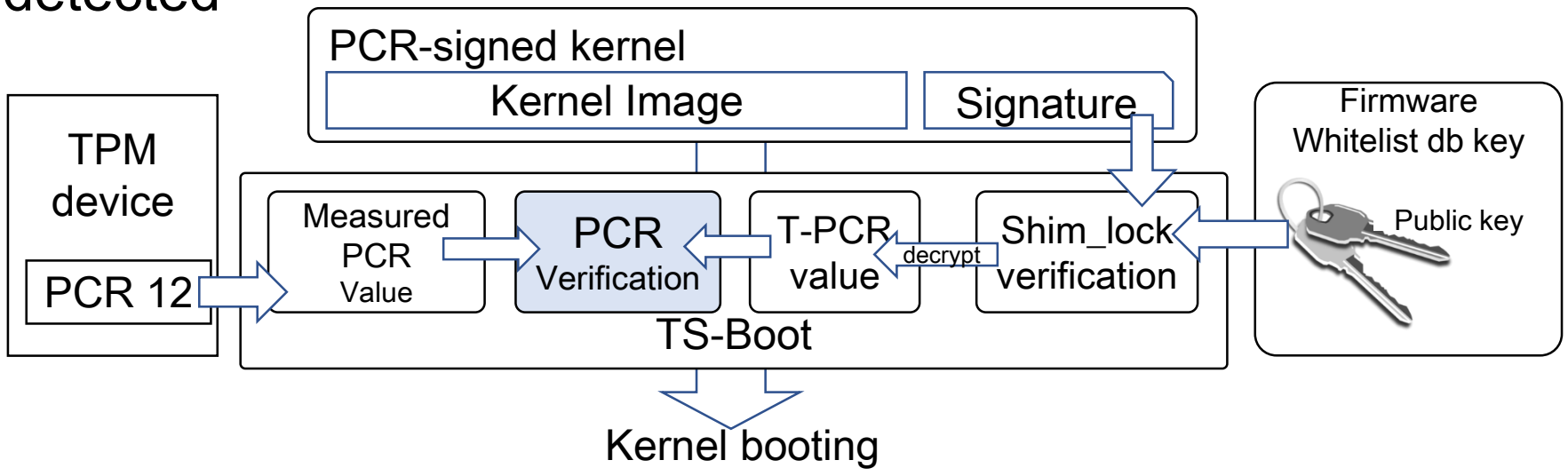
- Shim_lock verification

Thorough measurement / verification of the booting process

- Including Grub commands and modules
- **PCR-verification**

PCR-Verification

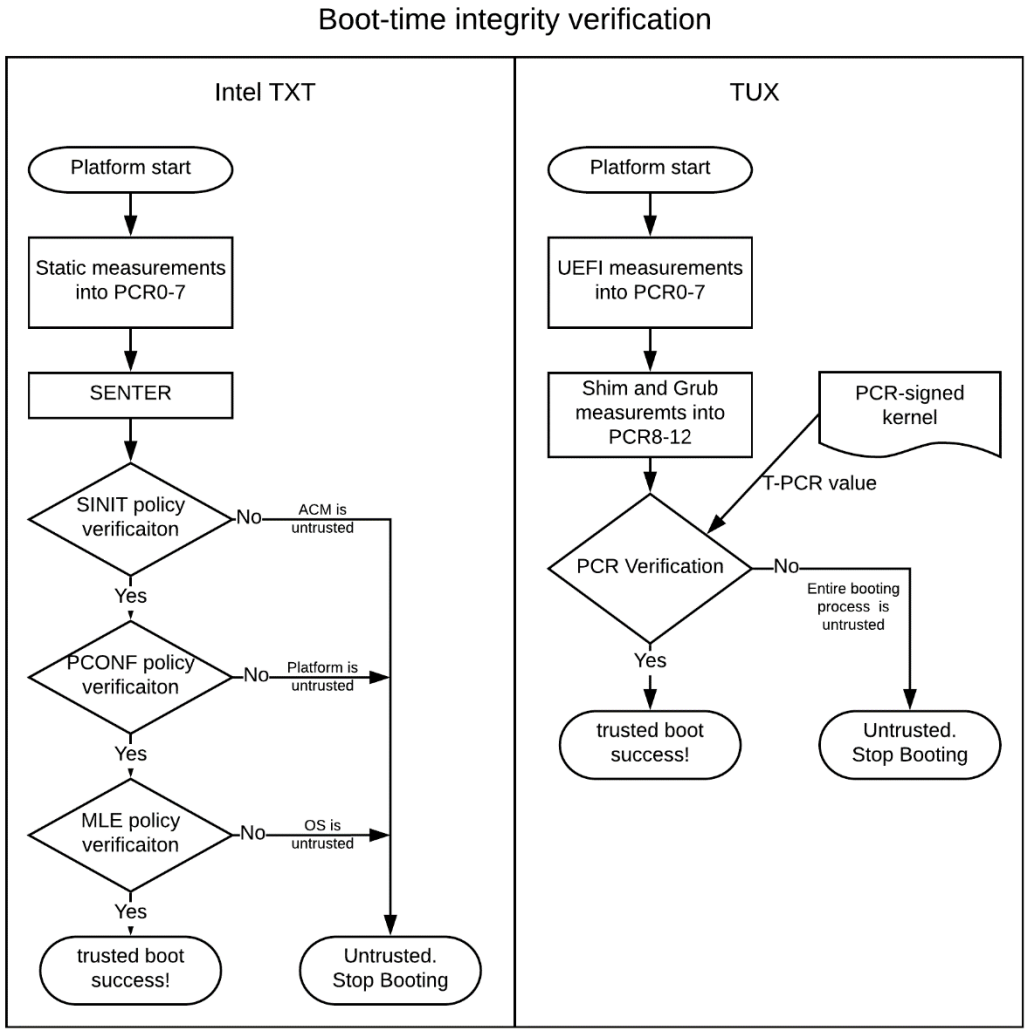
- Verifies the integrity of the entire booting process on runtime
- Extend measurement of every booting process using the PCR12
- Execute kernel using linuxefi function
  - Pass the kernel to Shim using shim_lock_verification
- Compare PCR12 with t-PCR stored in the kernel signature
- Any changes can be detected



Trusted Secure Boot (TS-Boot)

With TS-Boot, TUX…

Measure / verify the integrity of the entire booting process including Grub command and modules (Goal 3)



Experiment

- TPM measurement

A. PC1+SB on+Kernel 104	B. PC1+SB on+Kernel 109	C. PC1+SB off+Kernel 109	D. PC2+SB on+Kernel 109
Bank/Algorithm: TPM_ALG_SHA256(0x000b)			
PCR_00: 05 48 02 7e c	PCR_00: 05 48 02 7e c	PCR_00: 05 48 02 7e cf	PCR_00: 05 48 02 7e cf
PCR_01: f1 67 99 3b a	PCR_01: f1 67 99 3b a	PCR_01: f1 67 99 3b a5	PCR_01: c7 84 e6 09 94
PCR_02: 3d 45 8c fe 5	PCR_02: 3d 45 8c fe 5	PCR_02: 3d 45 8c fe 55	PCR_02: 3d 45 8c fe 55
PCR_03: 3d 45 8c fe 5	PCR_03: 3d 45 8c fe 5	PCR_03: 3d 45 8c fe 55	PCR_03: 3d 45 8c fe 55
PCR_04: f5 f8 1f 6b 5	PCR_04: f5 f8 1f 6b 5	PCR_04: f5 f8 1f 6b 5b	PCR_04: f5 f8 1f 6b 5b
PCR_05: de 89 35 69 c	PCR_05: de 89 35 69 c	PCR_05: de 89 35 69 c2	PCR_05: 39 55 01 58 89
PCR_06: 3d 45 8c fe 5	PCR_06: 3d 45 8c fe 5	PCR_06: 3d 45 8c fe 55	PCR_06: 3d 45 8c fe 55
PCR_07: 25 c0 b3 ce 4	PCR_07: 25 c0 b3 ce 4	PCR_07: 47 d9 c1 f4 d9	PCR_07: 25 c0 b3 ce 45
PCR_08: 63 81 11 5c d	PCR_08: f4 1e 86 df 9	PCR_08: b1 5d 09 67 39	PCR_08: a6 fe 12 0a 0f
PCR_09: e2 fa 1b a3 f	PCR_09: e2 fa 1b a3 f	PCR_09: e2 fa 1b a3 f9	PCR_09: e2 fa 1b a3 f9
PCR_10: 0b 74 50 53 8	PCR_10: 0b 74 50 53 8	PCR_10: 0b 74 50 53 8e	PCR_10: 0b 74 50 53 8e
PCR_11: 53 45 a7 13 8	PCR_11: 79 bd 24 78 8	PCR_11: 00 00 00 00 00	PCR_11: 79 bd 24 78 88
PCR_12: 92 5a 80 6e c	PCR_12: 31 68 59 c3 e	PCR_12: 10 5e fc 8c b1	PCR_12: 76 fc 4d 87 a9

TPM measurements

PCR #	Content	Measurement Host
PCR 0-7	BIOS and hardware configurations.	UEFI secure boot
PCR 8	Executed Grub commands.	Trusted Grub
PCR 9	Executed Modules from Trusted Grub.	Trusted Grub
PCR 10	Trusted Grub binary.	Shim
PCR 11	Kernel and initrd	Shim
PCR 12	Entire booting process.	UEFI Secure boot, Trusted Grub, and Shim.

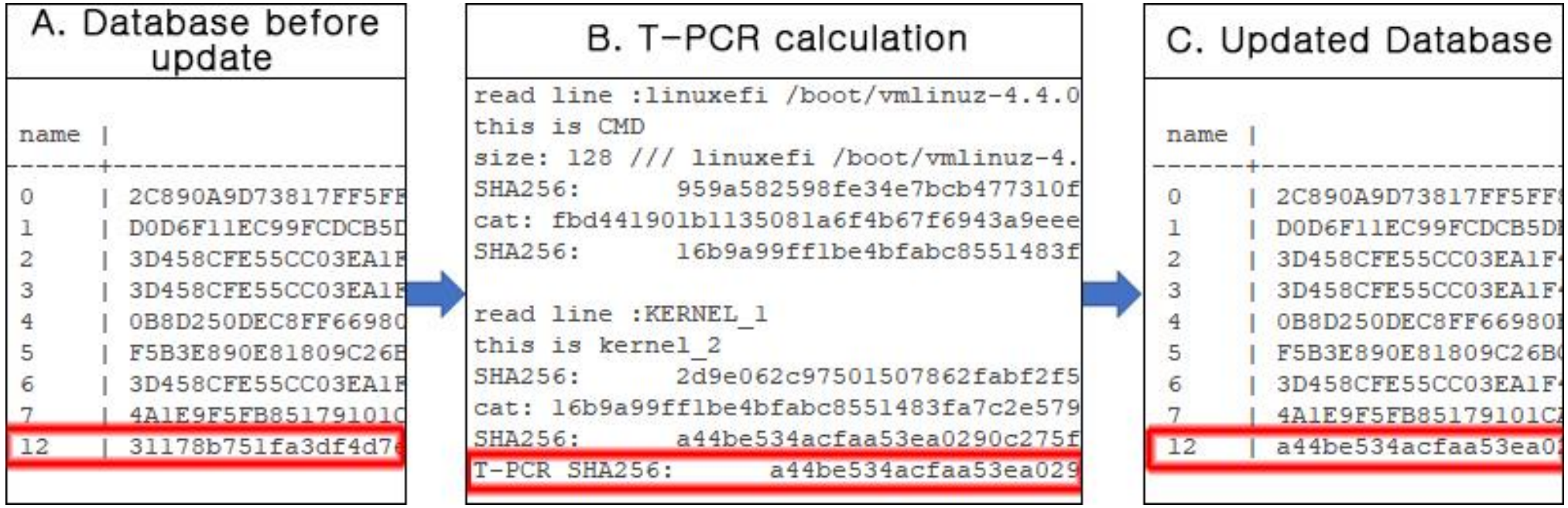
Experiment

- PCR-verification

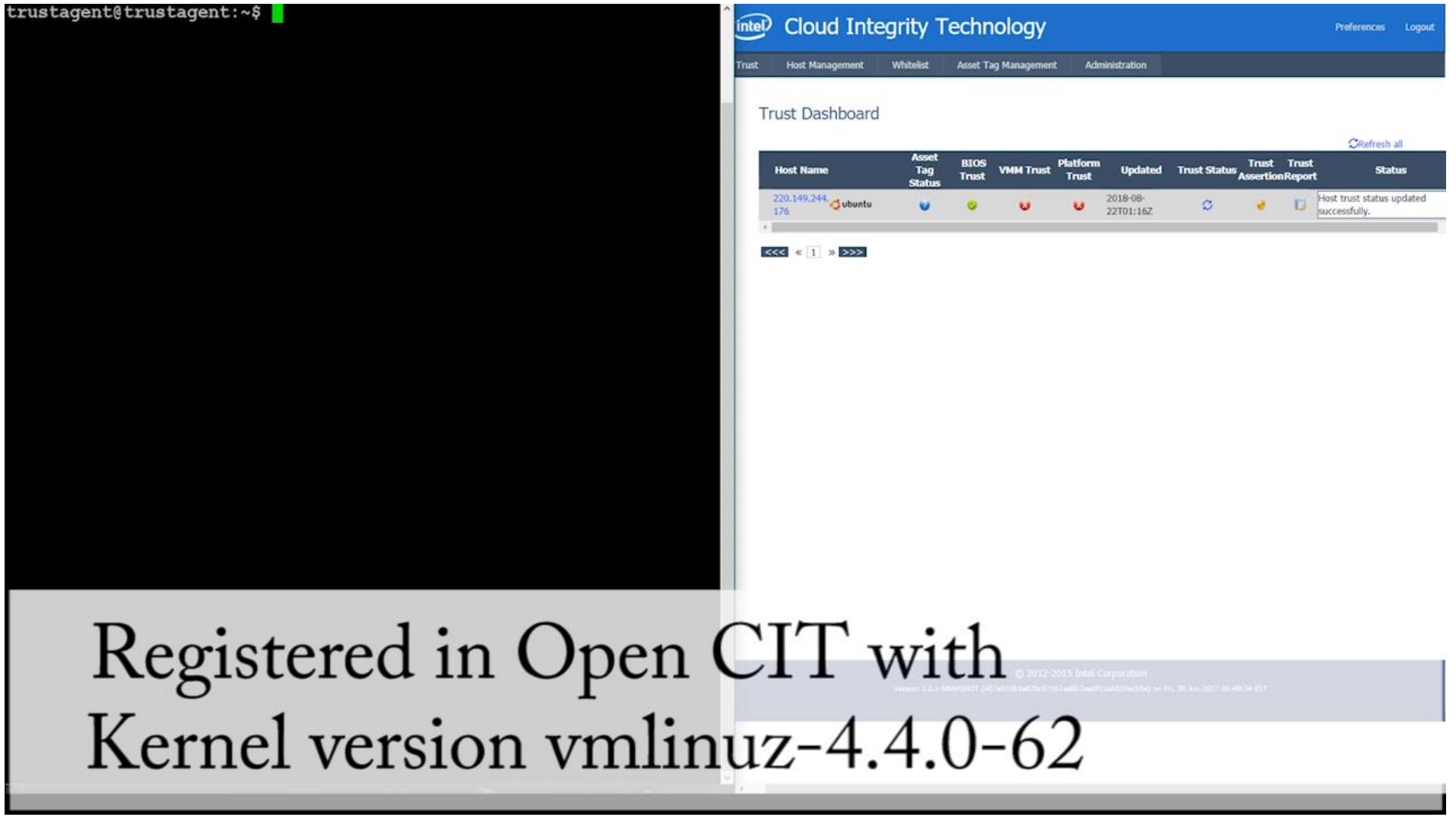
```
153             search --no-floppy --fs-uuid --set=roo
Inserted    fi
155         echo 'THIS IS TO FAIL PCR-VERIFICATION'
156         linuxefi                /boot/vmlinuz-4.4.0-104-
157         initrdefi               /boot/initrd.img-4.4.0-1
"grub.cfg" [Modified] 209 lines --45%--
```

Experiment

- whitelist update



Demo


trustagent@trustagent:~$
Cloud Integrity Technology
Preferences
Logout
Trust
Host Management
Whitelist
Asset Tag Management
Administration
Trust Dashboard
Refresh all
Host Name
Asset Tag Status
BIOS Trust
VMM Trust
Platform Trust
Updated
Trust Status
Trust Assertion
Trust Report
Status
220.149.244.176
ubuntu
2018-08-22T01:16Z
Host trust status updated successfully.
Registered in Open CIT with
Kernel version vmlinuz-4.4.0-62

Discussion

Roll-back and multiple kernel support

- Roll-back attack can be detected!

TUX owner's Key is Safe

- Open CIT server is safe / Firmware is safe

TUX may be applicable to environment other than desktop or server

SRTM or DRTM?

Conclusion

Integrity changes when update is conducted, and thus, it should be properly managed along with updates.

TUX...

- Extends Open CIT to transparently mange local updates
- Remote attestation with up-to-date whitelist
- Thorough integrity measurement, including Grub commands and modules
- Robust integrity verification with PCR-verification

Thank you.

Any Questions?