Software
Meets
Hardware
at
Semihalf

Improving security of FreeBSD with TPM 2.0 and Intel SGX

Kornel Dulęba

Presentation plan

- TPM overview
- Measured boot
- Strongswan with TPM
- SGX overview

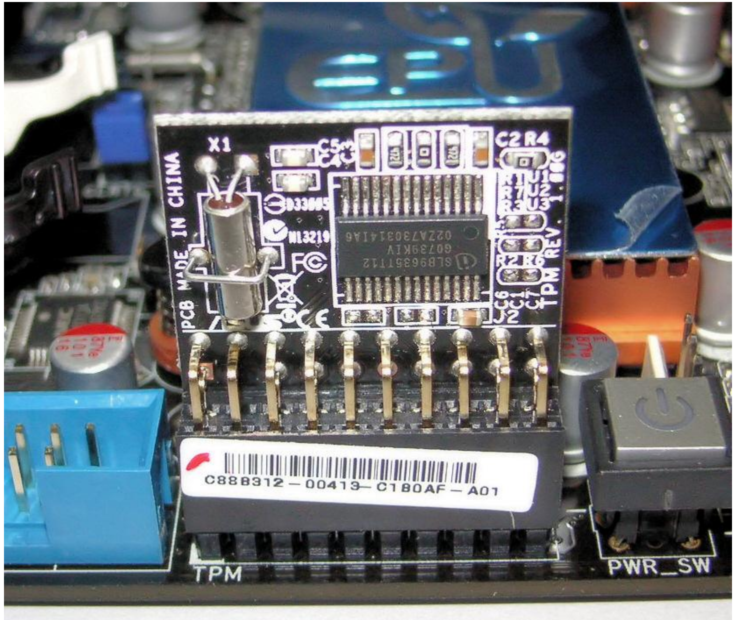
Presentation plan

- **TPM overview**
- Measured boot
- Strongswan with TPM
- SGX overview

HSM

- Secure storage for digital keys
- Onboard key generation
- Protection against physical tampering

TPM 101



TPM - $3.99 HSM

- Resources accessible through API
- Internal (d)RNG for key generation
- Anti hammering protection - makes dictionary attacks unfeasible
- Built to prevent physical tampering

Software Meets Hardware at Semihalf

Firmware TPM

- Must run in separation from the rest of system
- Much faster than discrete TPM - runs on main CPU
- Implemented in Intel ME, AMD PSP (check your BIOS)

Software Meets Hardware at Semihalf

TPM use cases

- Generating digital signatures, eg. SSH, IPSEC, 2FA, GPG
- Measured Boot
- MS Bitlocker / LUKS key storage (no GELI support yet)
- Secure storage of root certificates (integrity)

Software
Meets
Hardware
at
Semihalf

TPM authorization

- Hierarchies
- Basic passphrase based authentication
- Complex rules with Enhanced Authorization in TPM 2.0
- Can combine different assertions with AND/OR

Software Meets Hardware at Semihalf

TPM - EA policies

- Password
- HMAC (essentially PSK)
- PCR state (platform/boot integrity)
- Physical presence (press key, assert pin, access BIOS)
- Boot counter, internal clock

Software Meets Hardware at Semihalf

TPM caveats

- Poor software support
- Different pinout configurations
- Discrete chips are slow
- At least one successful, documented physical attack - Christopher Tarnovsky Black Hat 2010
- Are firmware implementations secure?

Software Meets Hardware at Semihalf

TPM in FreeBSD

- TPM 1.2 driver added in FreeBSD 8.2 (bsssd project)
- TPM 2.0 driver added by Semihalf in Dec 2018
- LPC bus only (no I2C/SPI support)
- Tested with Infineon SLB9665 TPM

Software Meets Hardware at Semihalf

Presentation plan

- TPM overview
- **Measured boot**
- Strongswan with TPM
- SGX overview

Software Meets Hardware at Semihalf

Measured Boot

- Store hashes of critical system components in PCRs
- Can only be read or extended
- Zeroed by TPM on reboot

$$PCR = hash(PCR|hash(Image))$$

Software Meets Hardware at Semihalf

Measured Boot

Firmware	Boot loader	OS
PCR0: UEFI image	**PCR4:** OS Loader (EFI apps)	**PCR8:** Kernel, etc*
PCR1: ACPI, SMBIOS	**PCR5:** Partition table(s)	
PCR2: Drivers from hdd	**PCR6:** Unused	
PCR3: EFI vars	**PCR7:** Secure Boot vars	

Source: TCG EFI Platform Specification

Software Meets Hardware at Semihalf

Event log

- UEFI creates entry in event log for every extend
- One can later compare the log entries against a database of trusted values
- Software can replay the extend operations and confirm log authenticity against signed PCR values (Quote operation)

Measured Boot

- Currently FreeBSD can't extend PCRs on its own
- UEFI measures every binary before passing execution to it - boot1.efi and loader.efi included
- Loader could be extended to measure kernel and modules too

Software Meets Hardware at Semihalf

Remote attestation

- Send event log together with signed PCR values
- Signature is generated using Quote operation
- Remote machine can now verify integrity of our firmware and possibly OS
- Works with Strongswan!
- On Linux only (Strongswan TNC bases on IMA)

Software Meets Hardware at Semihalf

CVE-2018-6622

- In S3/S4 power state TPM has no power.
- PCRs are supposed be preserved in NVRAM
- But TPM needs to be informed whether we are going to reset or sleep - by the OS

Software Meets Hardware at Semihalf

CVE-2018-6622

- Modify TPM driver, boot malicious OS, PCRs have wrong values
- Enter and immediately exit S3 state, PCRs are now resetted
- Spoof correct values
- Fixable in firmware

Presentation plan

- TPM overview
- Measured Boot
- **Strongswan with TPM**
- SGX overview

Strongswan

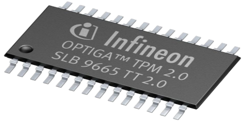
Strongswan

- Strongswan can use private keys stored inside a TPM
- Key is bound with a certificate used during IKE
- Access protected with a passphrase, either stored in clear text in configuration file or prompted for

Strongswan - IKE

Signing request with proper authentication

Encrypted digest



private key

TPM

Software
Meets
Hardware
at
Semihalf

Prerequisites

- FreeBSD TPM2.0 driver - available in 11.3
- TSS, a userspace library that can “talk” to the TPM

TSS

- Two libraries - one developed by Intel, other by IBM
- Intels library port in review on phabricator (17.09.19)
- IBMs follows POSIX, a one-liner patch in autoconf is needed for it to work

Caveats

- Although TPM has a key duplication mechanism (backup) it is undocumented and hard to use
- TSS still in development
- Little to no support on applications side
- Generating a signature using RSA2048 takes ~0.15s on Infineon SLB9665

Software Meets Hardware at Semihalf

Presentation plan

- TPM overview
- Measured Boot
- Strongswan with TPM
- **SGX overview**

Software Meets Hardware at Semihalf

Software Guard Extensions

- Developed by Intel for Skylake CPUs
- New special instructions
- Programs can be run in enclaves separated from each other and other parts of system
- Interrupts are supported,
works with OS scheduler (ERESUME)

SGX

- Enclave memory has to be reserved by firmware, it won't be accessible to the OS
- Not all BIOSes support it
- Memory management is done in OS driver

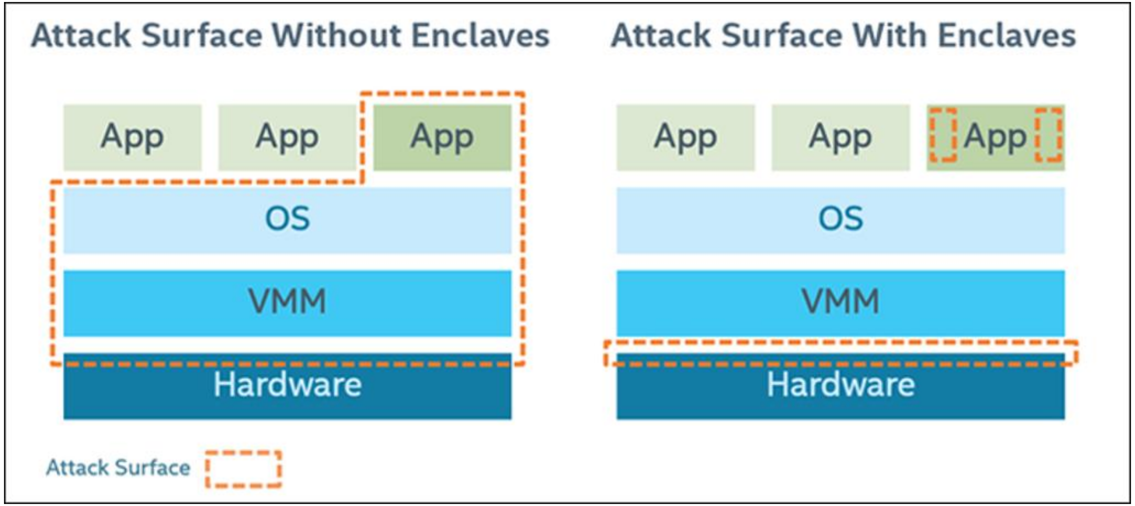
SGX

- Transparent encryption of both data and code
- Susceptible to side-channel attacks,
  data in CPU cache is not encrypted
- CVE-2018-3615 aka Foreshadow

SGX

- Cannot use syscalls, kernel is not trusted
- Libraries have to be linked statically against enclaves
- Intel provided a limited implementation of libc and a port of openssl

SGX


Attack Surface Without Enclaves
App
App
App
OS
VMM
Hardware
Attack Surface With Enclaves
App
App
App
OS
VMM
Hardware
Attack Surface

SGX

- All enclaves running in production mode have to be signed
- By default only Intel approved enclaves are allowed
- One can enter into a commercial agreement with Intel to have their certificate signed by them
- This enables Remote Attestation and prevents creation of undetectable malware

Software Meets Hardware at Semihalf

Launch Enclave

- Special enclave used to launch other enclaves
- After their signatures are verified
- The launch enclave itself is verified by the CPU
- Three MSRs containing SHA256 of public key used to sign Launch Enclave

Software Meets Hardware at Semihalf

SGX - FLC

- Flexible Launch Control
- Used to build own trust chain, independent from Intel
- MSRs with trusted hash can be overwritten
- FLC needs to be supported by both the firmware and CPU
- Fairly new feature, availability may vary

SGX - attestation

- Before sharing secrets we need to make sure that remote enclave is trusted
- Standard(ish) certificate chain attestation, with Intel issued CA
- CAs private key is derived from CPU fuses
- Can be used for DRM

Software Meets Hardware at Semihalf

SGX - sealing

- Enclaves don’t have any kind of non-volatile storage
- We need to preserve secrets outside of them in a secure way
- We don’t trust anyone - input integrity has to be validated
- AES-GCM scheme for sealing secrets

SGX - sealing

- Each CPU has some fuses burned at manufacturing time
- Sealing key is derived from the value of that fuse combined with enclave measurements
- As a result all secrets are bound to a CPU
- Simple interface, SDK handles key derivation

Software Meets Hardware at Semihalf

SGX - usages

- Offloading of cryptographic operations - similar concept to TPM
- Remote attestation - verify that application is executed on genuine Intel CPU in an enclave running known, trusted code
- Computation on remote, untrusted machine
- DRM - UHD Blu-Ray, Netflix 4k videos

SGX - Signal

- Privacy oriented communication app
- Checks who uses Signal basing on your contact list
- Send SHA256(phone number) for privacy
- Hashes can potentially be inverted (lookup table)
- Contact discovery server running in an enclave, clients can now verify its integrity using remote attestation

Source: https://signal.org/blog/private-contact-discovery/

Software Meets Hardware at Semihalf

SGX - Graphene

- “library OS” for running native Linux binaries in enclave
- Exposes glibc
- Built-in remote attestation support
- v1.0 released 11.09.2019

Software Meets Hardware at Semihalf

SGX in FreeBSD

- Kernel driver introduced in FreeBSD 12 by br@
- Partial port of Intel SGX SDK by br@
- It lacks some features, such as debugging
- Other libraries e.g SGX version of openssl were not ported

Acknowledgements

- Stormshield - initiators and sponsors of entire research and development

Software Meets Hardware at Semihalf

Questions?