Stato dell'arte su vulnerabilità e mitigazioni di

TRUSTZONE

Politecnico di Milano,
Advanced operating systems
A.A. 2021/2022

Docente: Vittorio Zaccaria
Studente: Daniele Carta

INTRODUZIONE

Politecnico di Milano,
Advanced operating systems
A.A. 2021/2022

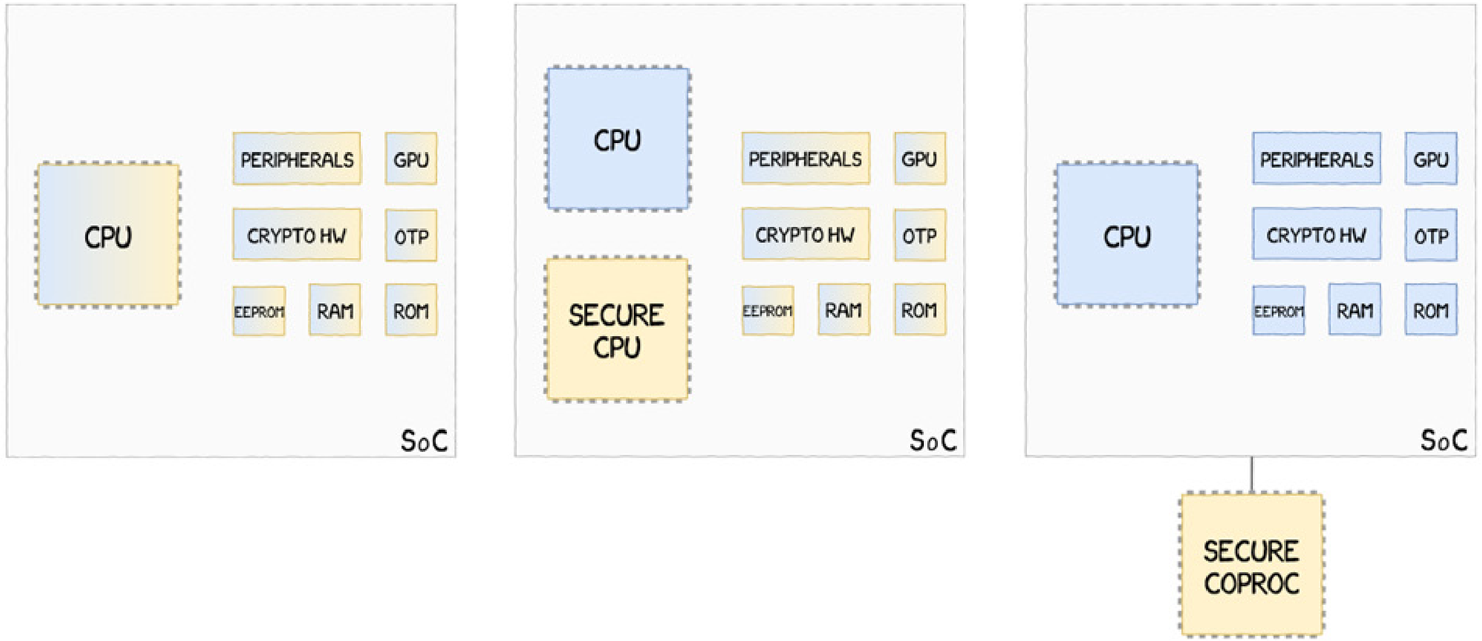
Docente: Vittorio Zaccaria
Studente: Daniele Carta

COS È TRUSTZONE?

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

TEE Enabler

Base di partenza per integrità e confidenzialità sull'esecuzione di codice in una parte del sistema



COS È TRUSTZONE?

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

Basata sull'hardware

NS bit

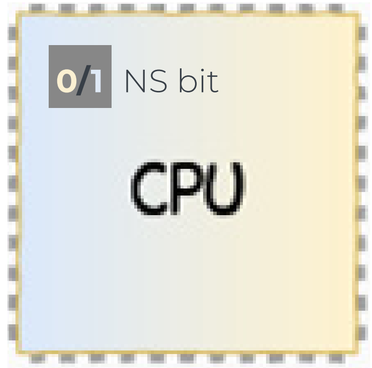
Può essere letto dal Secure World

Determina lo stato del processore che può essere sicuro o non sicuro

Consultando il **Secure Config Register**

Secure World

Non Secure World



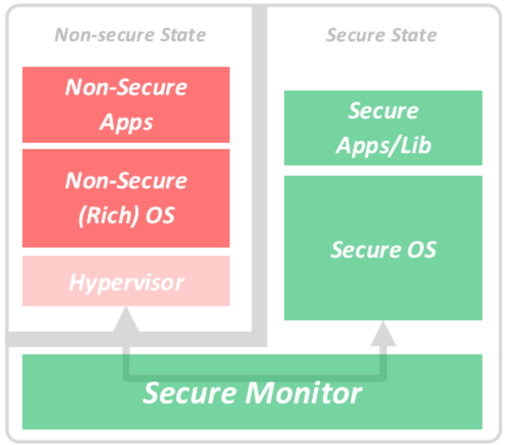
DUE MODELLI DI TRUSTZONE

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

CORTEX A

Utilizzo del Secure Monitor

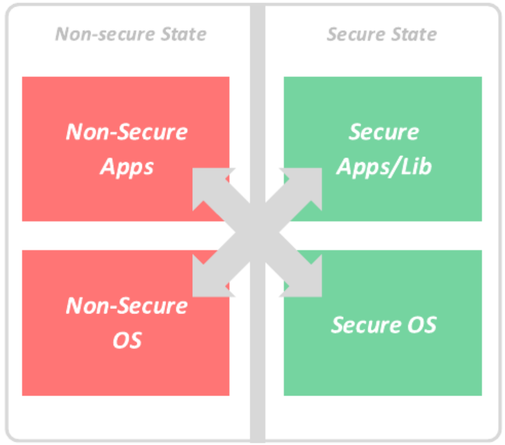
- Trusted kernel
- Può eseguire sistemi operativi



World switch

CORTEX M

Non utilizzo del Secure Monitor



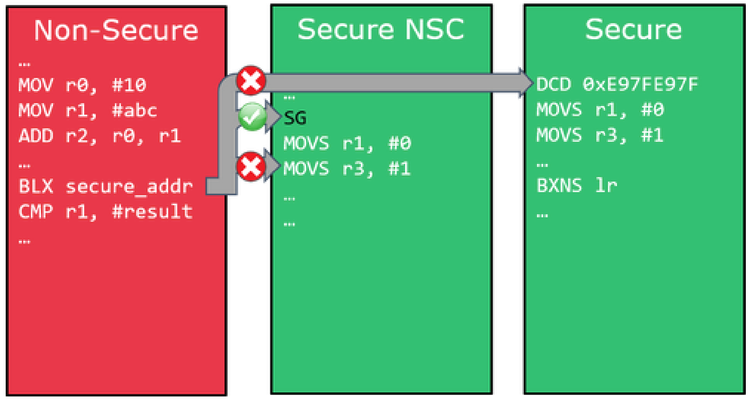
- Trusted service
- Context switch veloce
- Applicazioni per il basso consumo

CONTEXT SWITCH SU CORTEX-M

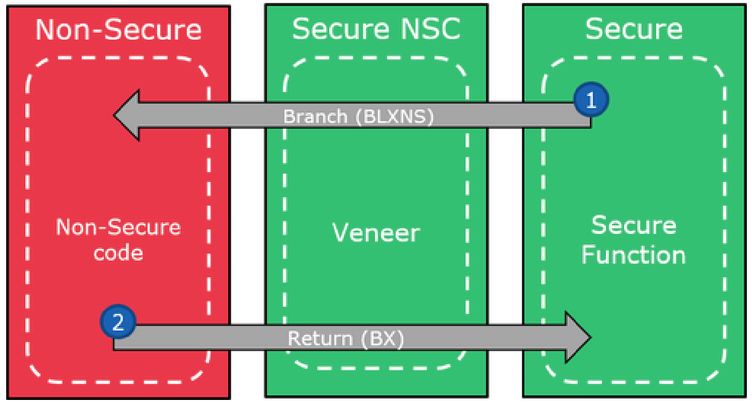
Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

3 istruzioni per gestire il context switch in maniera sicura:

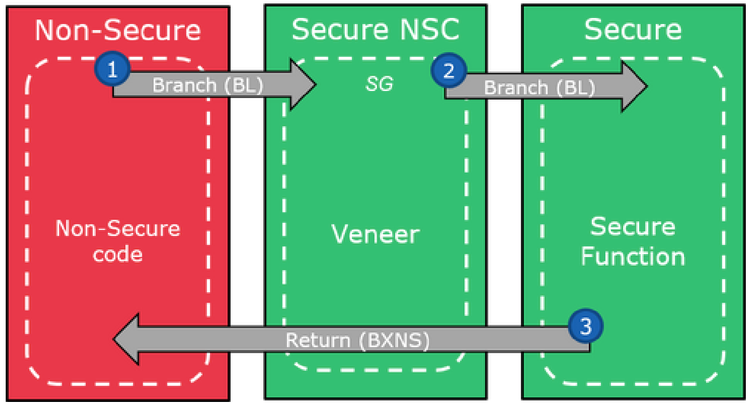
SecureGate:
Dal SW all'entrypoint di un istruzione del SW



Branch with eXchange to Nonsecure State:
Dal SW branch o return al NSW



Branch with Link and eXchange to Nonsecure State:
Dal SW chiama una funzione nel NSW



Il mapping della memoria è gestito da SAU e IDAU

CARATTERISTICHE DI TRUSTZONE E CONTROIDICAZIONI

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

- Piccola Trusted Code Base - API Standard dal 2009 a cura di GlobalPlatform	Spesso non rispettata dagli sviluppatori
- Estensibilità (Trustlets)	Problemi con le revoche
- Modularità (TZASC, TZMA, TPM...)	Hardware sensibile da configurare
- Isolamento hardware	Hardware condiviso

TRUSTED CODE BASE UN CONFRONTO

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

System	Core (bin / src)	TAs
Qualcomm TEE (Google Pixel XL)	1.61MB / –	2.71MB
Trustonic TEE (Samsung S7)	350KB / –	5,02MB
Huawei TEE (Huawei P8 Lite)	744KB /–	479KB
Nvidia TEE (Nvidia Tegra)	97KB / 123Kloc	80KB
Linaro TEE (Hikey960)	365KB /210Kloc	-
Linux (4.14.rc7)	19MB / 15Mloc	-
seL4 (kernel)	166.5KB / 19Kloc	-

Alle dimensioni del kernel si aggiungono quelle di:

- Trusted Applications
- Secure drivers
- Secure Monitor
- Firmware

VULNERABILITÀ

Politecnico di Milano,
Advanced operating systems
A.A. 2021/2022

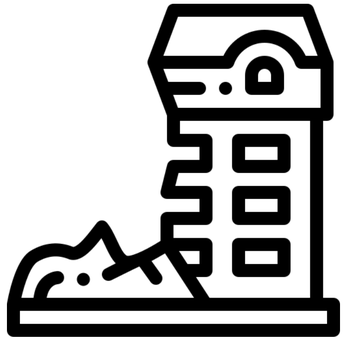
Docente: Vittorio Zaccaria
Studente: Daniele Carta

VULNERABILITÀ

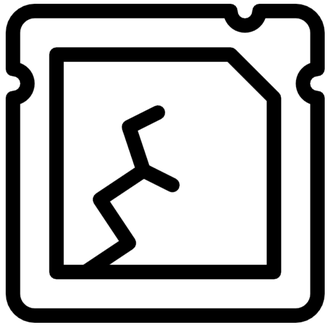
Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

IMPLEMENTAZIONE

ARCHITETTURA



HARDWARE



IMPLEMENTAZIONE

Politecnico di Milano,
Advanced operating systems
A.A. 2021/2022

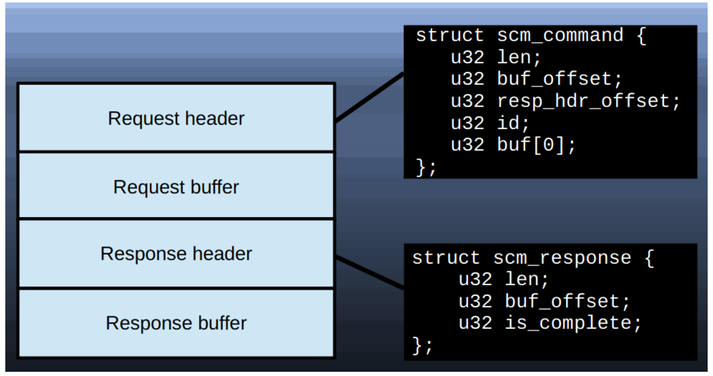
Docente: Vittorio Zaccaria
Studente: Daniele Carta

LA PRIMA VULNERABILITÀ NEL SM (BLACKHAT 2015)

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

Il TEE di Qualcomm usa un driver Linux per emettere SMC attraverso due metodi → Secure Channel Manager

- Call by registers
- Request/Response Structure → Attraverso un'area di memoria condivisa



LA PRIMA VULNERABILITÀ NEL SM (BLACKHAT 2015)

Controllo sull'area di memoria condivisa

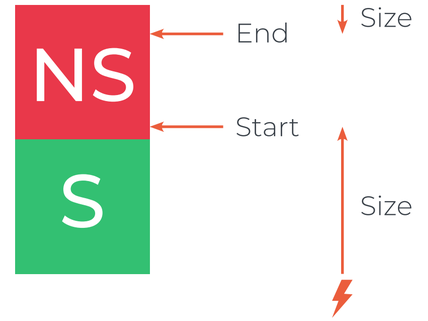
Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

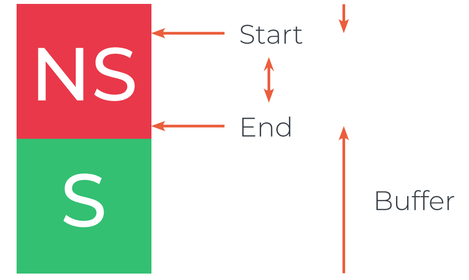
Risiede nel mondo non sicuro?

↓

Il TEE scriverà in quel buffer la sua risposta...

```
int qsee_is_ns_memory(long addr, long size)
{
return qsee_range_not_in_region(qsee_region_list, addr, addr+sizeaddr+size);
}
int qsee_range_not_in_region(void *region_list, long start, long end)
{
long tmp;
if (end < start) {if (end < start) {
tmp = start;tmp = start;
start = end;start = end;
end = tmp;end = tmp;
}}
/* Validate that start to end doesn't overlap
* secure list */
...
}
```





I SOLITI BUG

Class	Subclass	# Bugs
Validation Bugs	Secure Monitor	2 (1.07%)
	Trusted Applications	62 (33.16%)
	Trusted Kernel	52 (27.81%)
	Secure Boot Loader	5 (2.67%)
Functional Bugs	Memory Protection	32 (17.11%)
	Peripheral Configuration	8 (4.28%)
	Security Mechanisms	11 (5.88%)
Extrinsic Bugs	Concurrency Bugs	11 (5.88%)
	Software Side Channels	4 (2.14%)

→ Corruzione della memoria

→ Più bug nel bootloader che nel SM

→ Timing side-channel attacks

ARCHITETTURA

Politecnico di Milano,
Advanced operating systems
A.A. 2021/2022

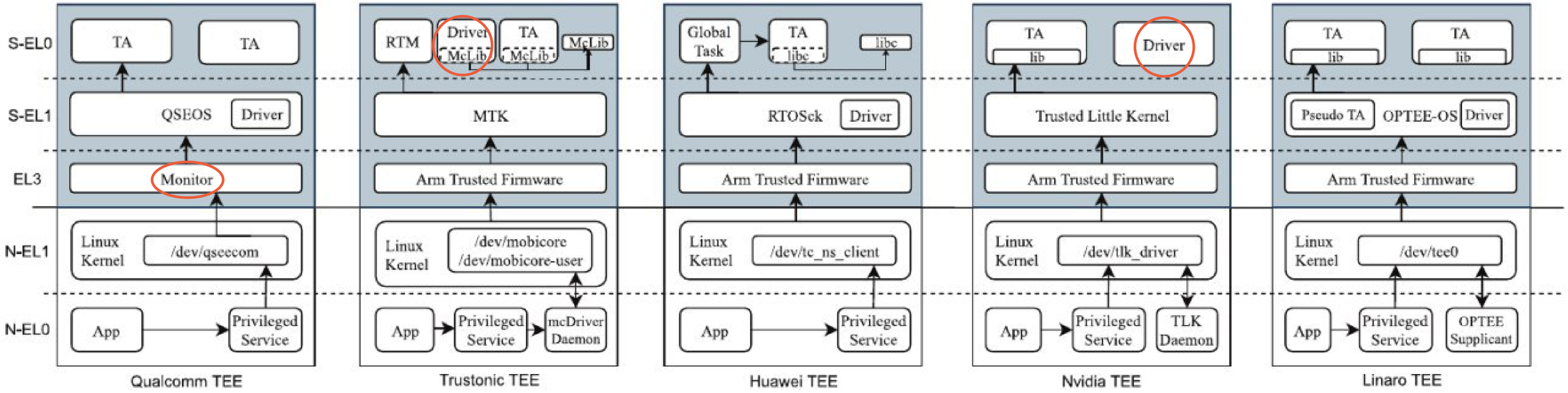
Docente: Vittorio Zaccaria
Studente: Daniele Carta

DRIVER

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

- Sono complessi e fonte di bug
- Solo alcuni vivono nel kernel space
- Altri nelle mani delle TAs

Architettura di ciascuna implementazione



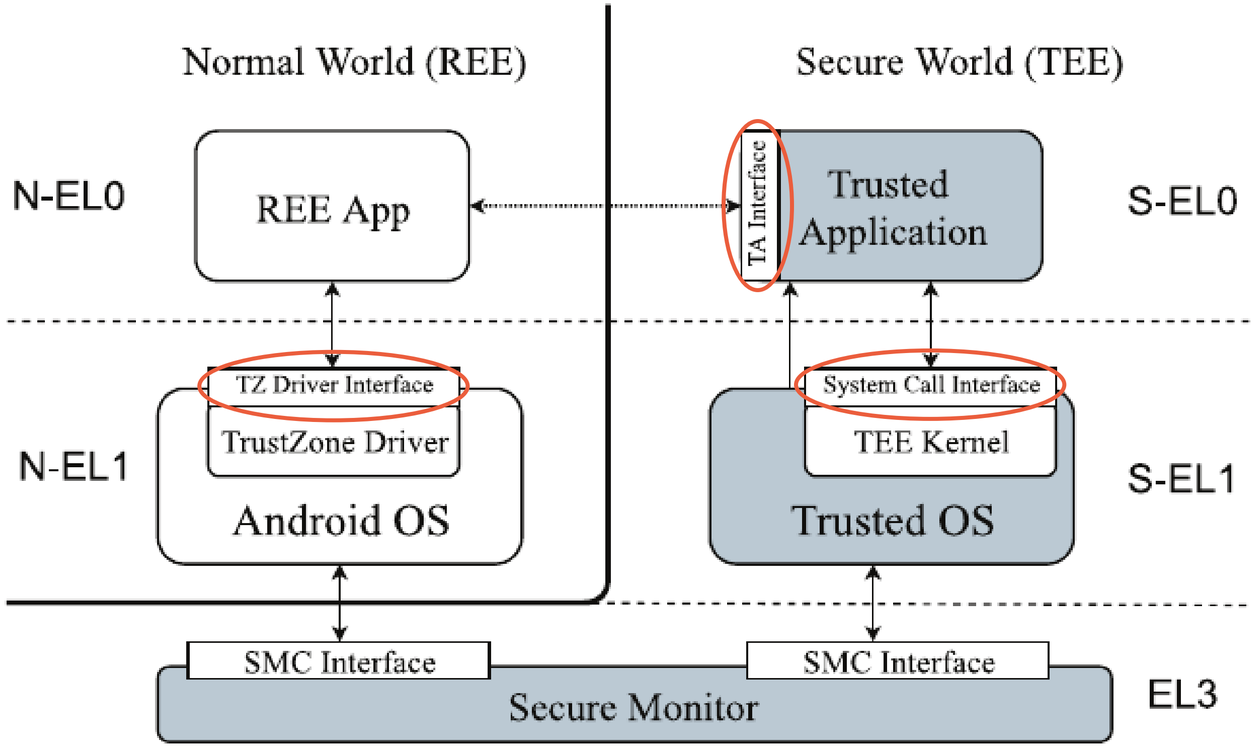
INTERFACCE

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

- In Android 4 daemons hanno accesso privilegiato al driver di TrustZone

- La TA Widivine implementa 70 comandi

- QSEE offre 69 diverse syscalls contro le 116 di Linux



BREAKING SAMSUNG'S TRUSTZONE (BLACKHAT 2019)

Target: trusted application del TEE di Trustonic

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

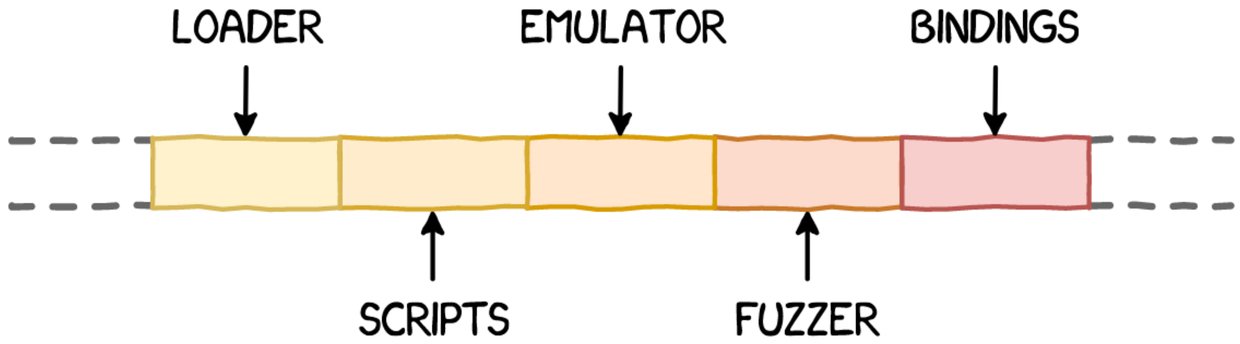
STEP #1 - Caricamento su IDA/Ghidra

STEP #2 - Identificazione delle funzioni

STEP #3 - Creare un emulatore

STEP #4 - Cercare vulnerabilità con un Fuzzer

STEP #5 - Exploitare sul vero target



BREAKING SAMSUNG'S TRUSTZONE (BLACKHAT 2019)

Cosa hanno trovato?

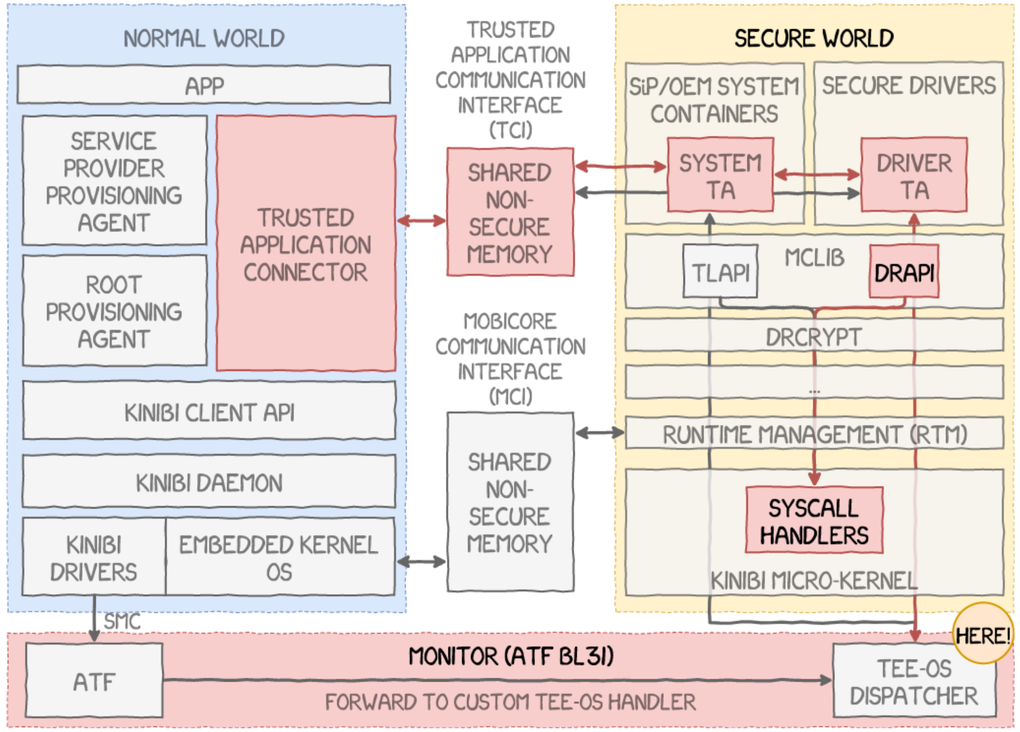
Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

Buffer Overflow da manuale

- Nelle TAs
- Nei Secure Driver

↓

Esecuzione di codice arbitrario nel Secure Monitor



ISOLAMENTO TRA SW E NSW

Boomerang attack

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

Le TAs possono mappare la memoria fisica del NSW

↓

Da una TA posso compromettere il non trusted kernel

↓

Trasformando TrustZone in un vettore d'attacco

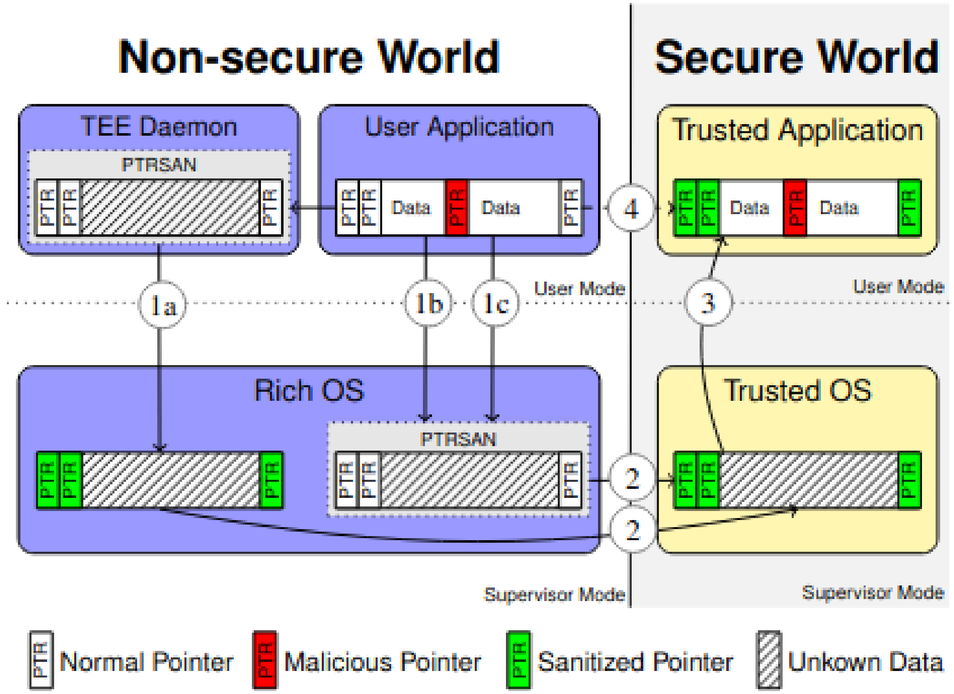


Fig. 2: An example of BOOMERANG, where a malicious memory pointer is hidden from pointer sanitization, ultimately tricking a TA to act on that memory address.

REVOCA DELLE TA

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

-TA caricata dal filesystem dell’untrusted OS e verificata dal trusted OS

-Caricamento di una nuova versione della TA

-Prevenzione del caricamento della versione obsoleta e vulnerabile

REVOCA DELLE TA

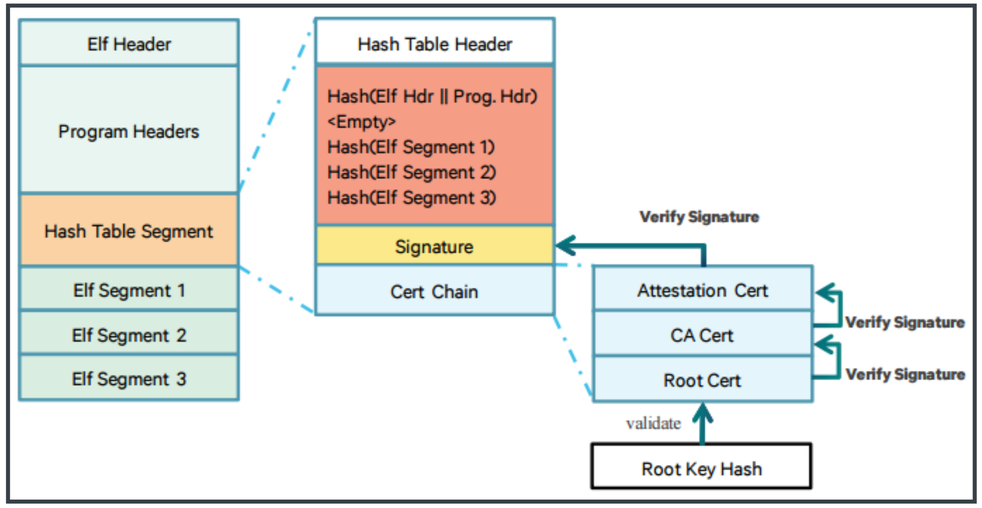
Project Zero's Trust Issues (blogpost 2017)

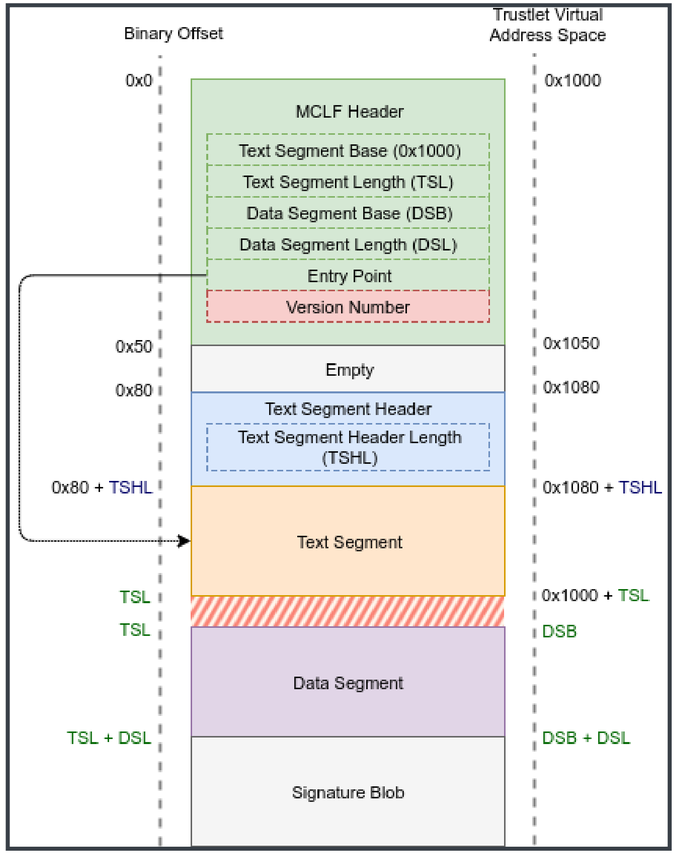
Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

-Buona documentazione che spiega il procedimento di revoca

-Signature sulla TA computata sulla base di un version ID

-Tutti gli ID di tutti I trustlet di Qualcomm e Trustonic sono '0'





HARDWARE

Politecnico di Milano,
Advanced operating systems
A.A. 2021/2022

Docente: Vittorio Zaccaria
Studente: Daniele Carta

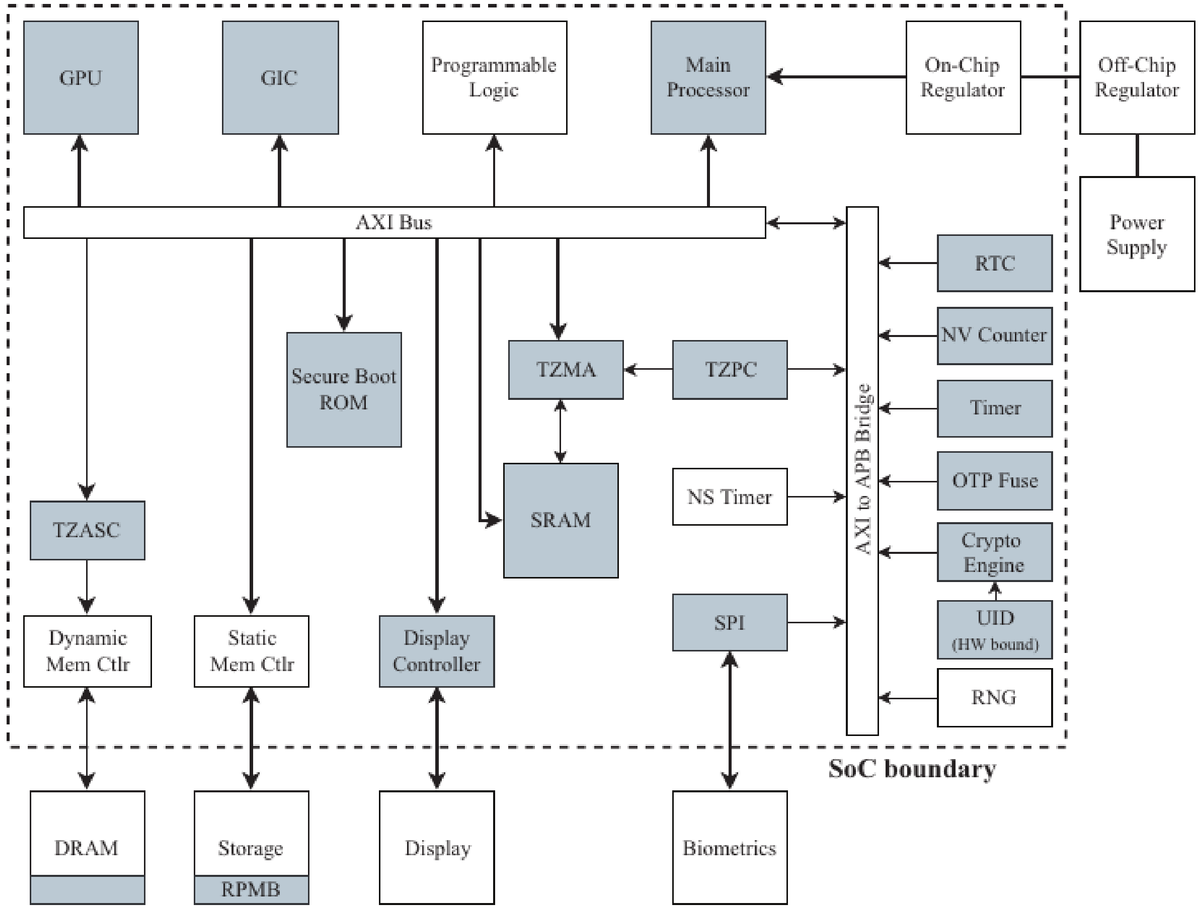
HARDWARE

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

-TCB estesa al firmware

-Configurazione delicata

-FPGA



Componenti sicuri
Componenti NS

CHAIN OF TRUST

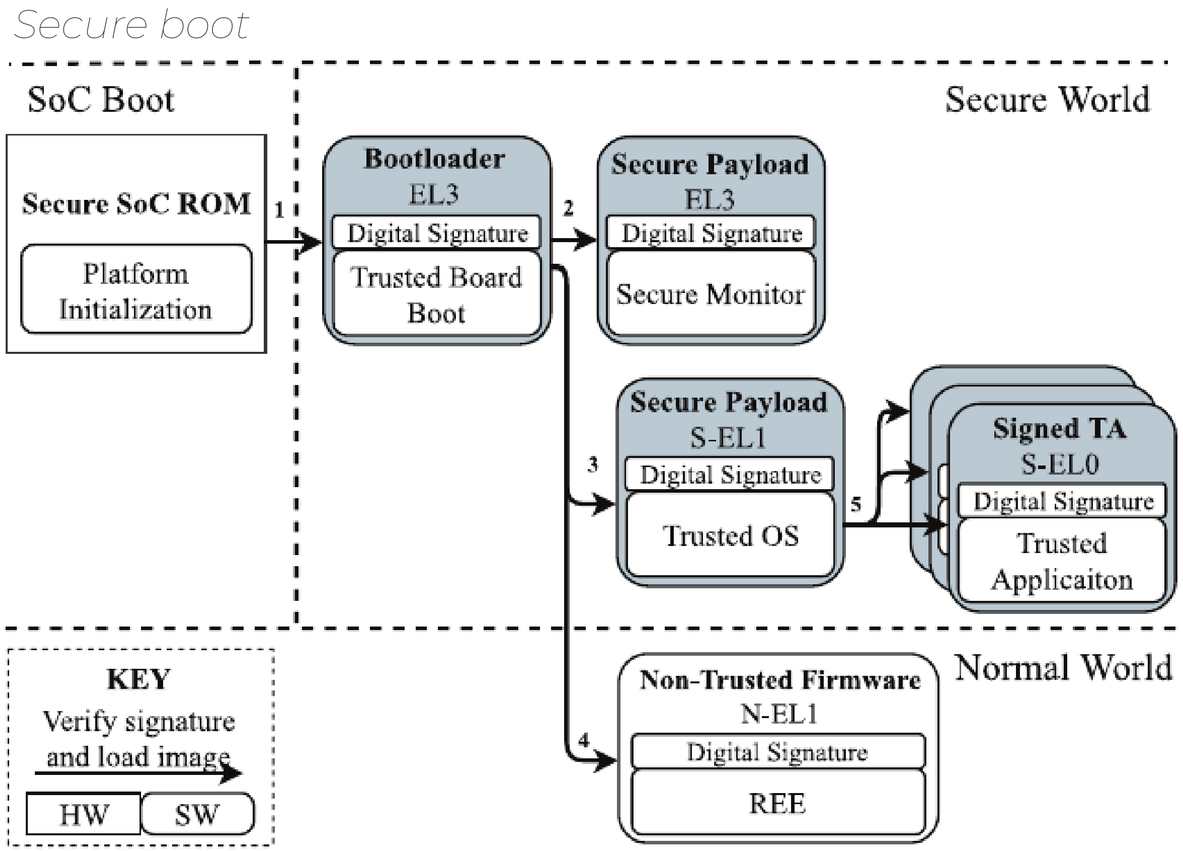
Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

-Affidabilità della chain of trust

-TOCTOU

-Microprobing

-Silicon editing



POWER MANAGEMENT

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

Ad uso del NSW

↓

Fault injection via userspace software

↓

CLKSCREW

Possibilità di configurare l'hardware per lavorare su limiti di frequenza maggiori di quelli che dovrebbero essere imposti dal venditore

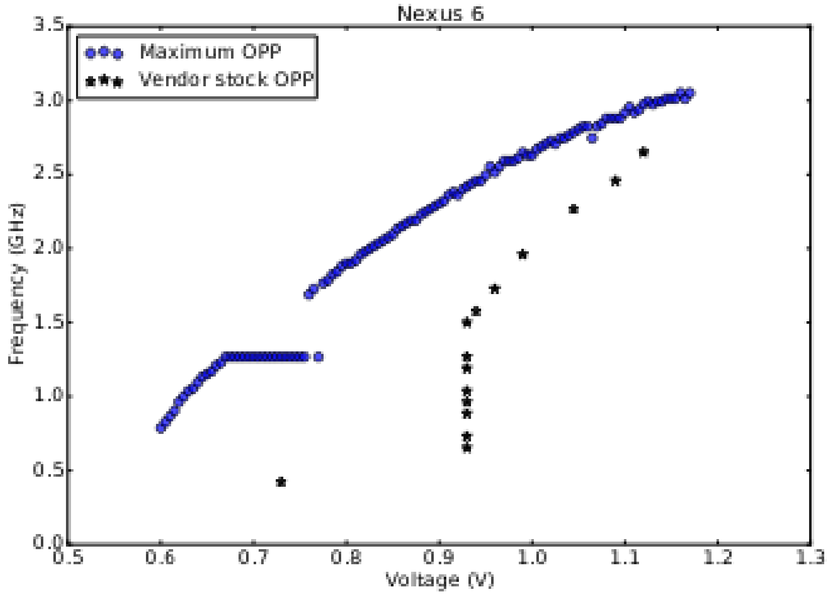


Figure 5: Vendor-stipulated voltage/frequency Operating Performance Points (OPPs) *vs.* maximum OPPs achieved before computation fails.

POWER MANAGEMENT

Fault injection (36c3 2019)

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

Chip SAML11 di microchip (TrustZone-M)

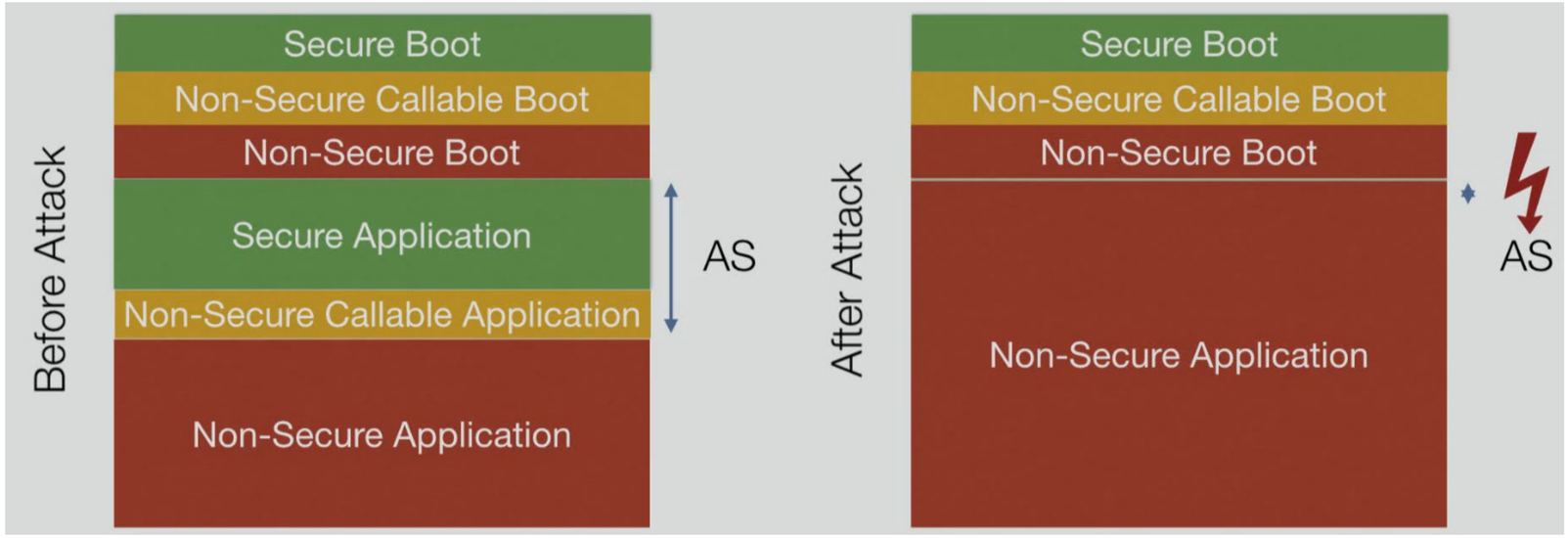
Non ha SAU

IDAU configurata dalla boot rom

Trovato attraverso power analysis

Setting di registro AS

Glitch oltre il setting di AS

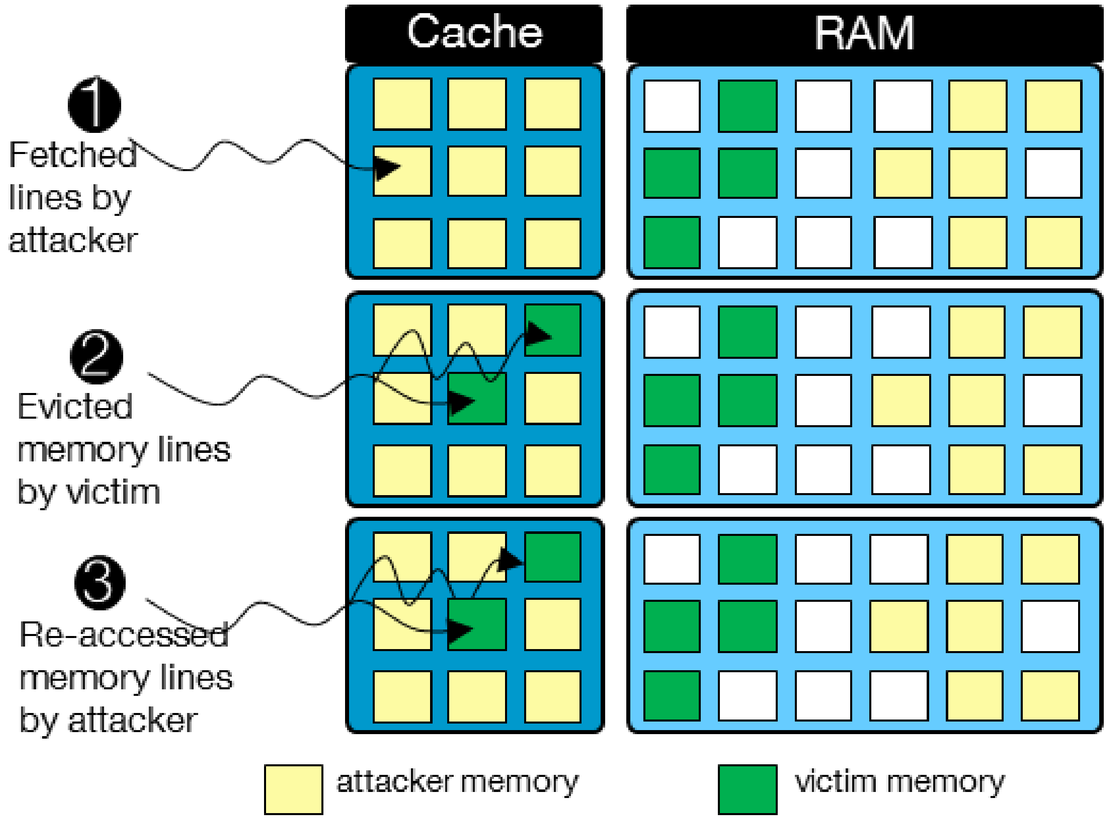


MICROARCHITETTURE

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

-Rowhammer e Branch predictor leaks

-Cache condivise → Prime & Probe



MICROARCHITETTURE

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

Migliorando Prime&Probe (34c3 2017)

- Sfruttare un secondo core e interrupt → - Migliorare granularità temporale

- Usare i performance counter → - Ridurre il rumore

MITIGAZIONI

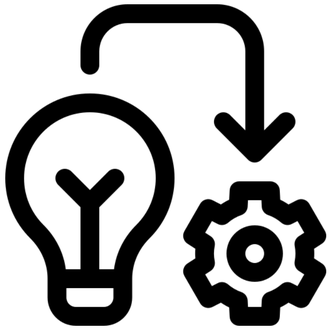
Politecnico di Milano,
Advanced operating systems
A.A. 2021/2022

Docente: Vittorio Zaccaria
Studente: Daniele Carta

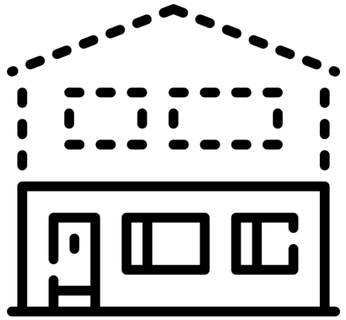
MITIGAZIONI

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

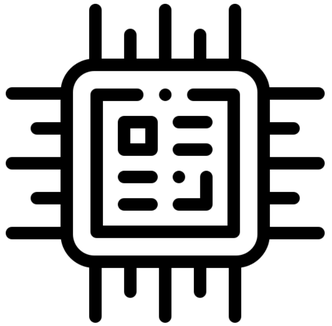
IMPLEMENTAZIONE



ARCHITETTURA



HARDWARE



MITIGAZIONI IMPLEMENTAZIONE

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

- Utilizzo di linguaggi type e memory safe

- RustZone è un estensione per OP-TEE per lo sviluppo di TAs con Rust

→ Validazione e concorrenza

- Verifica del software via metodi formali
  Complesso ma fattibile (MIPE, Komodo)

→ Funzionali

MITIGAZIONI ARCHITETTURA

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

- Rispettare l’architettura

- Aumentare la granularità dell’isolamento

- Canali di comunicazione fra SW e NSW sicuri (SeCReT)

- Memoria criptata come in Intel SGX

- Contenere la grandezza della TCB

MITIGAZIONI HARDWARE

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

- Eliminare la shared cache
- Fornirne una dedicata al SW
- Flushare ad ogni utilizzo del SW

→ Cache

- Limitare il controllo di energia a livello driver o hardware

→ Power management

- Un AXI per soli dispositivi sicuri

→ Componenti

- Per la chain of trust Physical Unclonable Functions
- Remote attestation, TPM

→ Attestazione

PROTEZIONE DELLA MEMORIA

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

- ASLR assente o debole

- Niente canaries, guardpages, NX bit

Mechanisms		Qualcomm	Trustonic	Huawei	Nvidia	Linaro
User	ASLR	◐	○	○	○	○
Space	SC	●	○	○	○	○
	GP	○	○	–	–	–
	XP	WXN	WXN	○	UXN/PXN	UXP/PXN
Kernel	KASLR	○	○	○	○	○
Space	SC	●	○	○	○	○
	XP	WXN	WXN	○	UXN/PXN	UXN/PXN

Non trovato

CONCLUSIONI

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

- Trade off con performance e costo
- Industria molto frammentata
- SoC e board sovraffollati
- Focus sulla root e la chain of trust
- La sicurezza come vettore d’attacco
- Tema di ricerca molto attiva

BIBLIOGRAFIA

Politecnico di Milano
Advanced operating systems 2021-2022
Daniele Carta

- SoK: Understanding the Prevailing Security Vulnerabilities in TrustZone-assisted TEE Systems, David Cerdeira et al.
- Demystifying Arm TrustZone: A Comprehensive Survey, Ssndro Pinto, Nuno Santos
- How to Break Secure Boot on FPGA SoCs through Malicious Hardware, Nisha Jacob et al.
- BOOMERANG: Exploiting the Semantic Gap in Trusted Execution Environments, Aravind Machiry et al.
- CLKSCREW: Exposing the Perils of Security-Oblivious Energy Management, Adrian Tang et al.
- SeCReT: Secure Channel between Rich Execution Environment and Trusted Execution Environment, Jinsoo Jang et al.
- Reflections on Trusting TrustZone - Blackhat 2015, Dan Rosenberg
- Breaking Samsung’s ARM trustzone - Blackhat 2019, Quarkslab
- Trust Issues: Exploiting TrustZone TEEs , Project Zero
- Microarchitectural Attacks on Trusted Execution Environments - 34c3, Ryan Keegan
- TrustZone-M(eh): Breaking ARMv8-M’s security - 36c3, Thomas Roth
- RustZone: Writing Trusted Applications in Rust - Blackhat 2018, Eric Evenchi
- Breaking Samsung’s Root of Trust: Exploiting Samsung S10 Secure Boot

GRAZIE

Politecnico di Milano,
Advanced operating systems
A.A. 2021/2022

Docente: Vittorio Zaccaria
Studente: Daniele Carta