ræelize

Trust Ain't Easy:
Challenges of TEE Security

Niek Timmers
niek@raelize.com
@tieknimmers

Cristofaro Mune
cristofaro@raelize.com
@pulsoid

Overview

- Introduction
- Challenges for TEEs
- Conclusion
- Q&A

Introduction

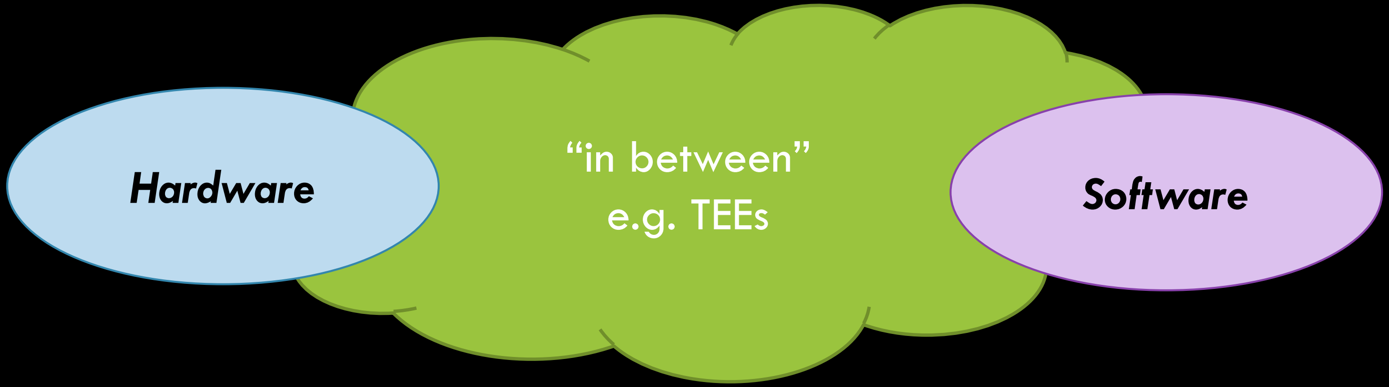
raelize

Cristofaro Mune

- Co-Founder at Raelize
- ~15 years experience analyzing and testing the security of complex systems and devices

Niek Timmers

- Co-Founder at Raelize
- ~10 years experience analyzing the security of devices

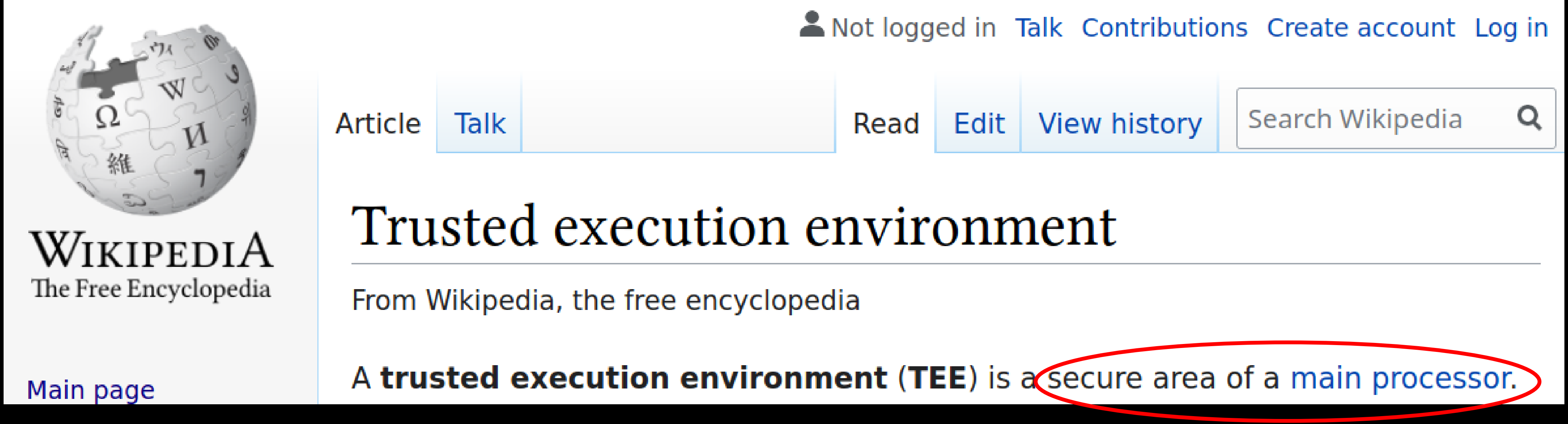


We've been analyzing and testing TEEs for ~10 years

Incorrect perspective.

Definition?

- A TEE is often believed to be a 'processor feature'



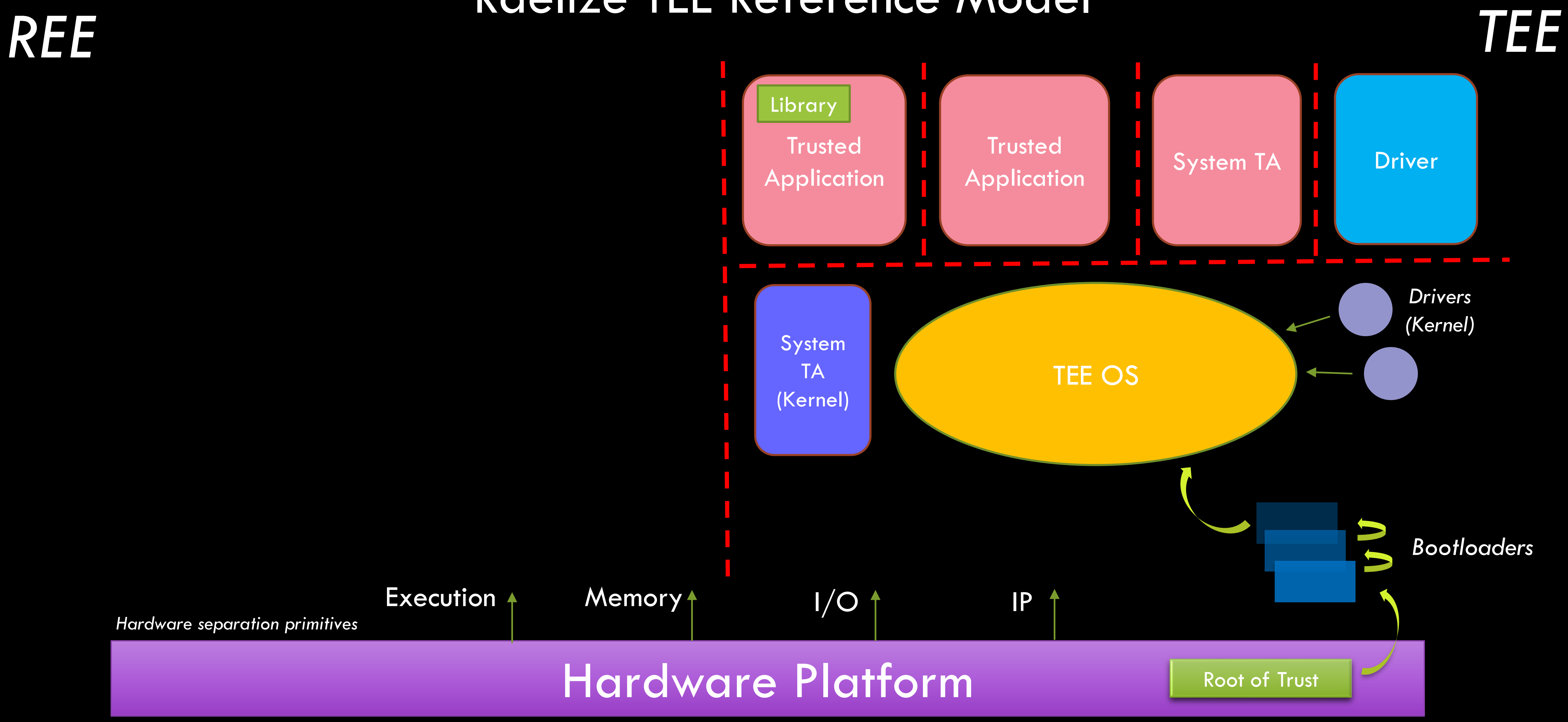
Trusted execution environment

From Wikipedia, the free encyclopedia

A **trusted execution environment** (**TEE**) is a secure area of a main processor.

This is mostly incorrect.

Raelize TEE Reference Model

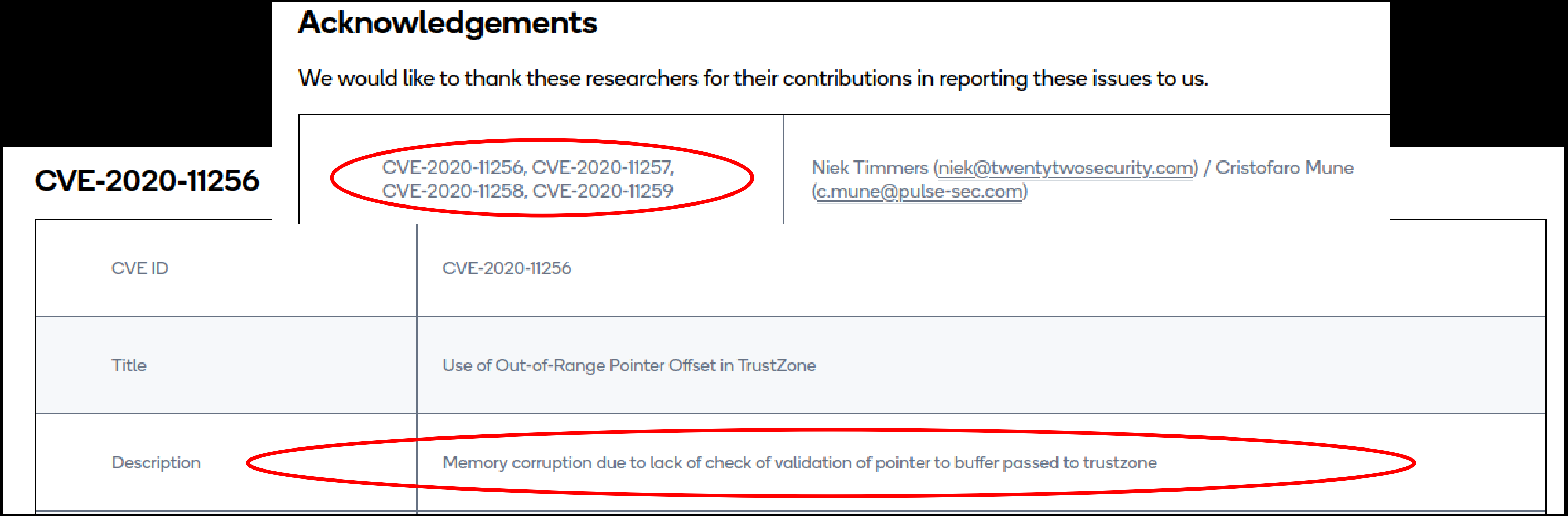


Actually…

- Separations are fundamental for a TEE
  - Memory
  - Hardware modules (i.e. IP)
- Separations are enforced by hardware controllers
  - Memory Protection Unit (MPU)
  - TrustZone Address Space Controller (TZASC)
  - TrustZone Protection Controller (TZPC)
  - …

Pointers are historically causing headaches…
(e.g. memory addresses)

Qualcomm QSEE vulnerabilities



Acknowledgements

We would like to thank these researchers for their contributions in reporting these issues to us.

CVE-2020-11256, CVE-2020-11257, CVE-2020-11258, CVE-2020-11259	Niek Timmers (niek@twentytwosecurity.com) / Cristofaro Mune (c.mune@pulse-sec.com)

CVE-2020-11256

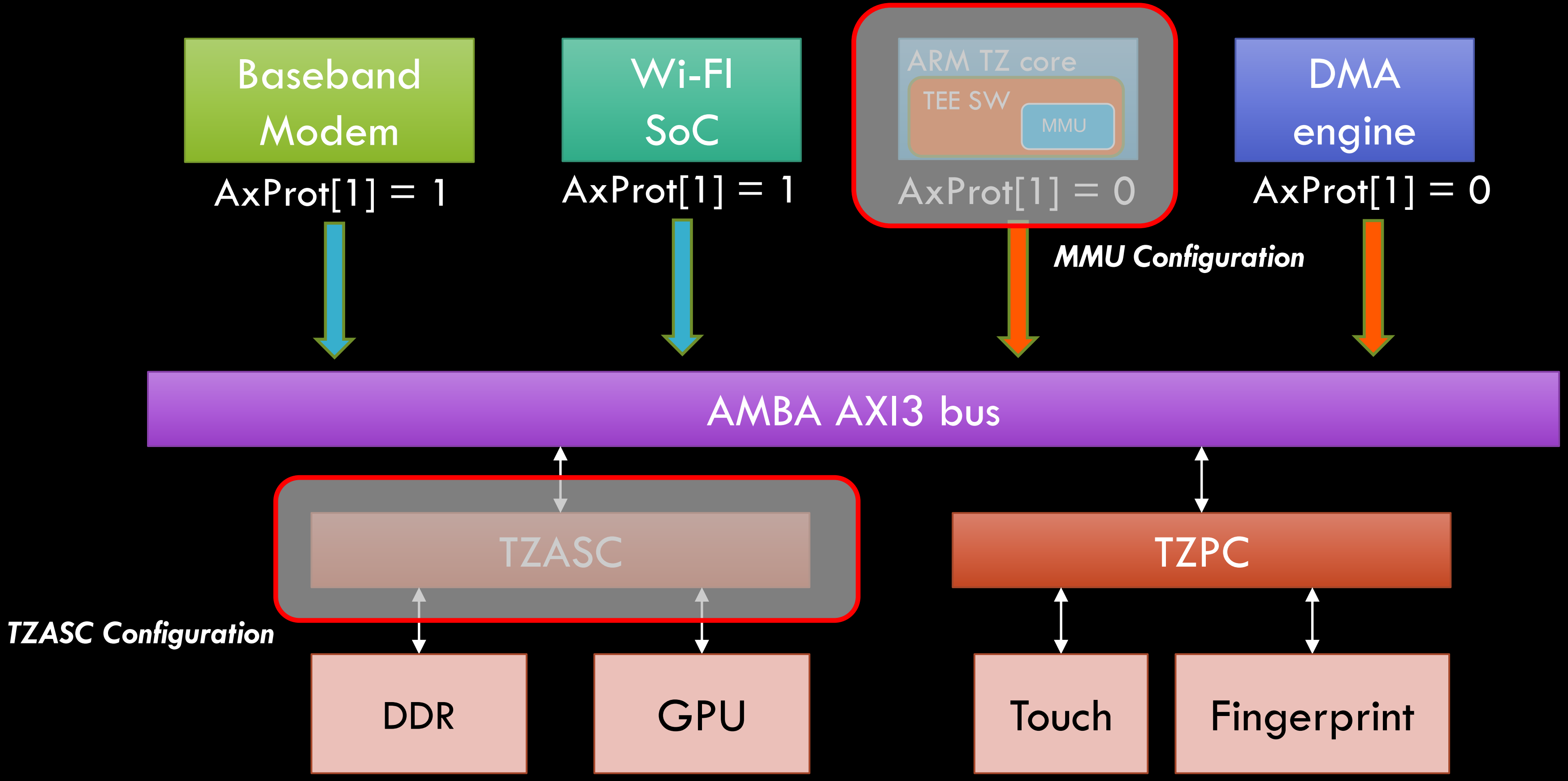
CVE ID	CVE-2020-11256
Title	Use of Out-of-Range Pointer Offset in TrustZone
Description	Memory corruption due to lack of check of validation of pointer to buffer passed to trustzone

Source: Qualcomm Security Bulletin (January 2021)

Unchecked pointers leading to TEE code execution

Consistency is challenging.

Secure Memory: MMU and Controllers views



Independent. Unrelated.

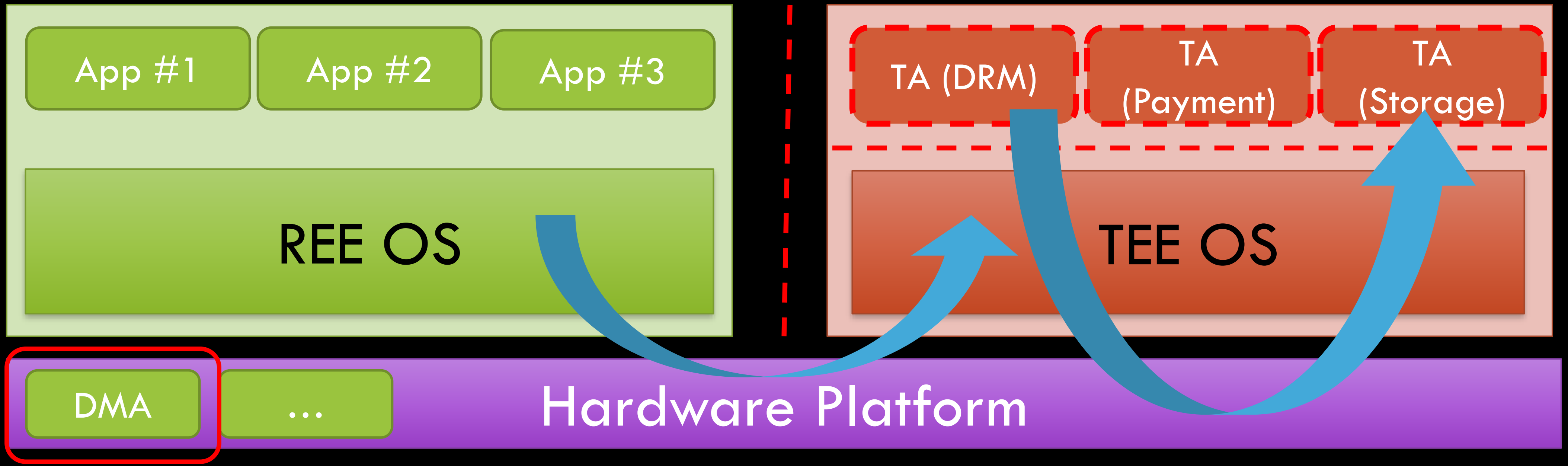
Fragmented view of secure memory

- No system-level view of (secure) memory
- Information spread across many configurations
  - TrustZone controllers, MMU, MPU, …
  - Secure range configuration in software (i.e. tables)
- No dedicated functionality to determine what’s REE or TEE memory

Threat modeling is hard.

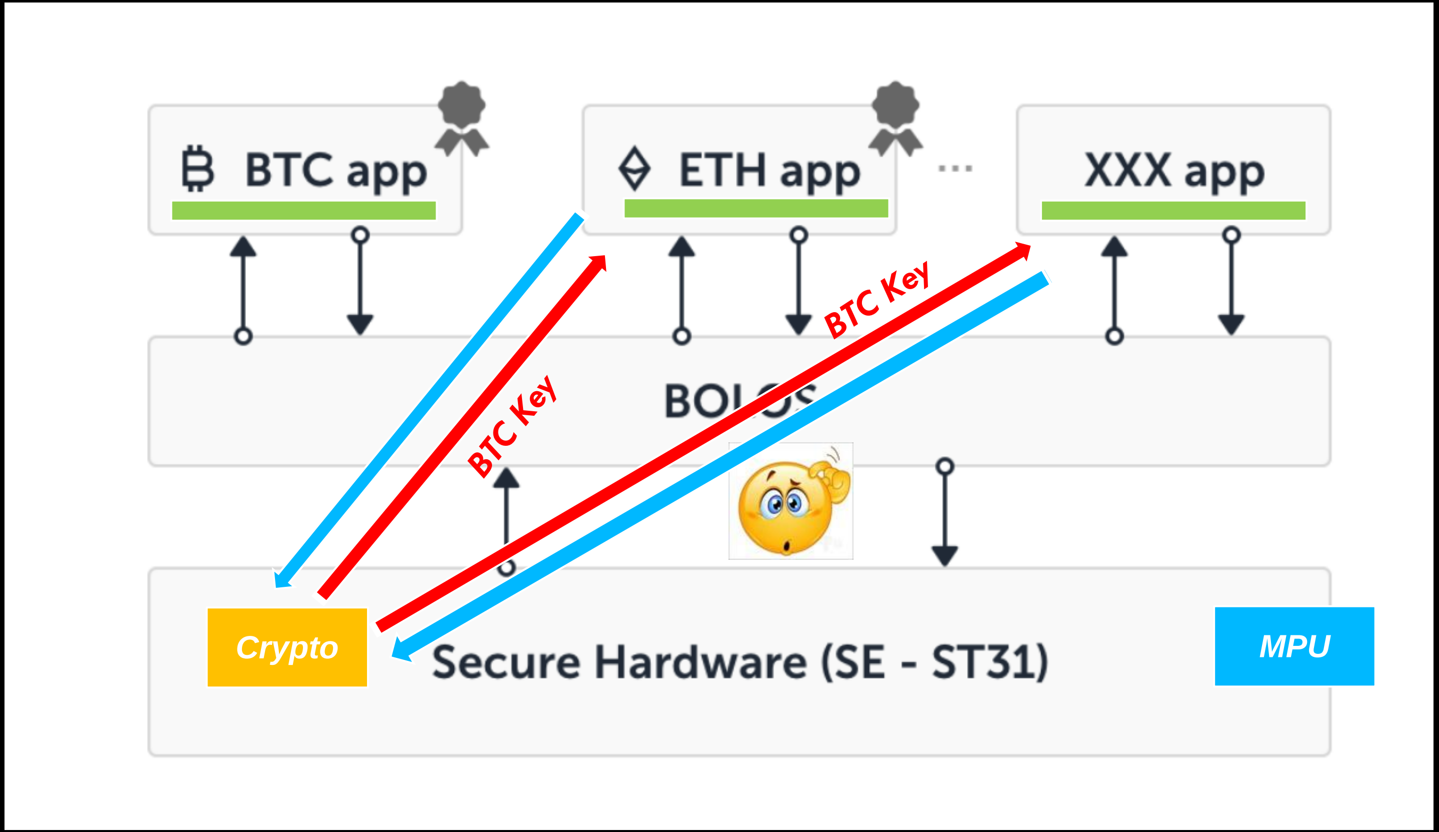
Using hardware to cross boundaries

- Design may let hardware IPs unrestricted access to memory



- Use DMA-capable engines to access across boundaries

Ledger Nano Crypto Wallet



HW IP separation for TAs is often overlooked

Technology aint’t easy.

Availability is not enough.

- ARMv8.3 pointer authentication
  - Great, but slow adoption…
- Software exploitation mitigations (i.e. ASLR, W^X, canaries, etc.)
  - Common in REEs; but less for TEEs…
- Also… are security features (e.g. Secure Boot) really secure?

Technology has limitations

- Not all platforms support advanced security features
  - E.g. No pointer authentication on ARMv7, ARMv8-M, etc.

- Some security features are not effective in restricted environments
  - E.g. ASLR implementations in a TEE may enjoy little entropy

Complexity is significant.

Configuration can be challenging

- Securely configuring a TEE is not trivial
  - Controllers, HW modules, registers, memory layout,…
- Dynamic configuration by multiple components
  - Personalization, bootloaders, operating system, etc.
- Maintenance required across product releases

Diverse ecosystem

- Devices are not made by a single entity (e.g. company)
  - E.g. SoC manufacturer is not the developer of the TEE OS
- Multiple entities with different responsibilities
  - E.g. SoC manufacturer is not responsible for configuring the TEE securely
- Inconsistencies at boundaries yield opportunities for attacks
  - E.g. boundary between components

Product certification is sub-optimal.

Certification

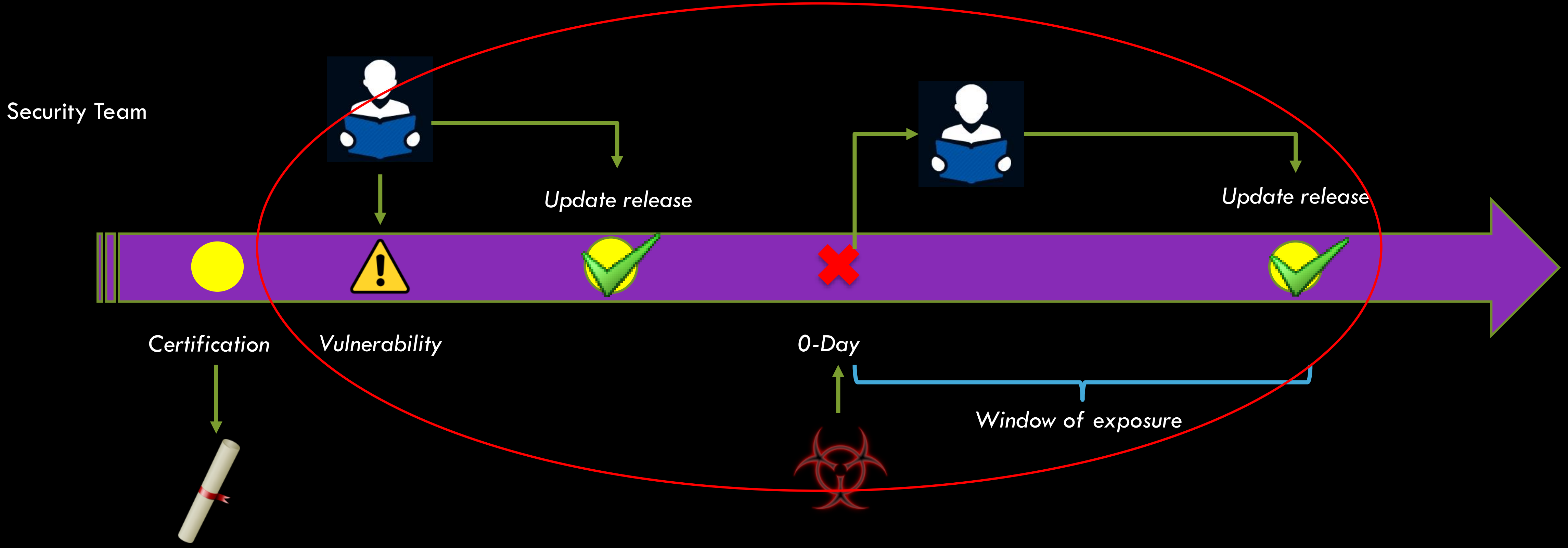
- Works well for hardware (immutable)
  - Once evaluated, it will not change anymore

- Works less for software (mutable)
  - Software is dynamic in nature (i.e. updates, etc.)
  - Code base size of a TEE is often large

Do you prefer a certified product or a secure product?

All products are vulnerable… security reduces risks.

In other industries…



Keeping products secure is key…

Provoking thoughts

- Handling security incidents should be the new "NORMAL"
  - This needs a well-defined process

- Why don't we evaluate and certify THAT process?

- Certifying companies vs certifying (only) their products

Does your organization have a security contact?

The bright side…

Positive developments #1

- New technology is available
  - Actively developed operating systems (i.e. OP-TEE, Trustonic, etc.)
  - Hardware partitioning (i.e. ARM v8.4+)
  - Security hardening features (i.e. ARM v8.3+)

- Check the presentations at LVC2021 on these topics!

Positive developments #2

- Interfaces are (being) standardized
  - ARM Trusted Firmware (i.e. TF-A, TF-M)
  - ARM Platform Security Architecture (PSA) Firmware Framework
  - GlobalPlatform API specification

- Having a proper security posture is becoming more widespread
  - Security contact
  - Collaboration with researchers
  - “Vulnerability reward programs” (aka “Bug bounties”)

Let’s wrap up.

Conclusions

- Thorough understanding of a TEE is key for securing it
- Available technology should be used as intended
- Processes should be certified, not only products
- Important lessons can be learned from other industries

Before we end...

Want to find out more?

ræelize

Training Week

April 19, 2021 - April 22, 2021

Click for more info!

More details about our research:
https://raelize.com/blog

TEEPwn

Breaking TEEs by Experience

BootPwn

Breaking Secure Boot by Experience

rælize

Thank you! Any questions!?

Niek Timmers
niek@raelize.com
@tieknimmers

Cristofaro Mune
cristofaro@raelize.com
@pulsoid