UNIVERSITY OF CAMBRIDGE

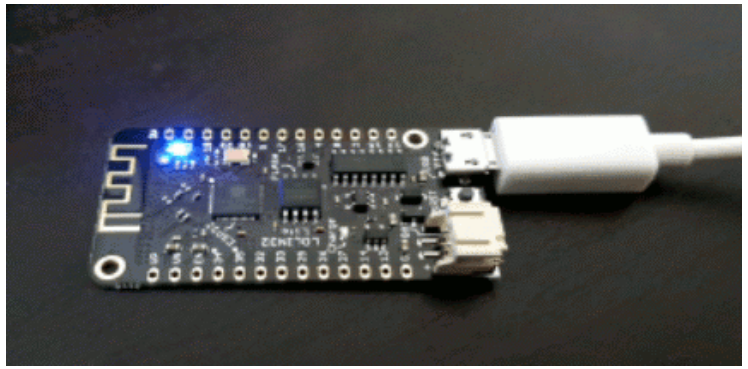
Towards Tiny Trustworthy Enclaves for Unikernels

Zahra Tarkhani, Anil Madhavapeddy

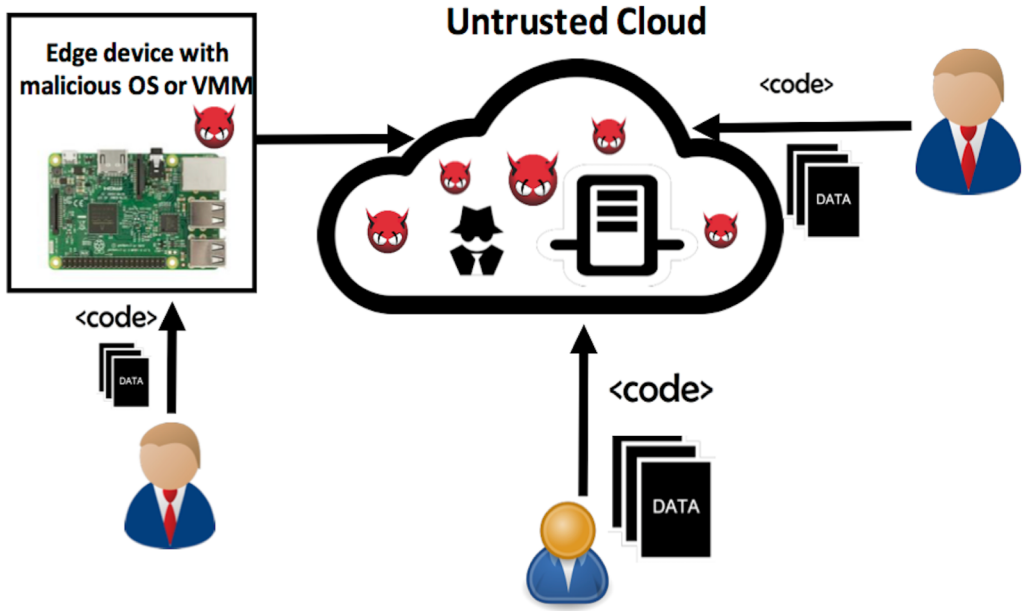
(email: first.last@cl.cam.ac.uk)

Motivation

- IoT is transforming our daily life
- IoT applications major requirements
  - Scalability, low latency, resource efficiency
  - IoT edge/cloud security and privacy concerns are **REAL**
  - Unikernels can help

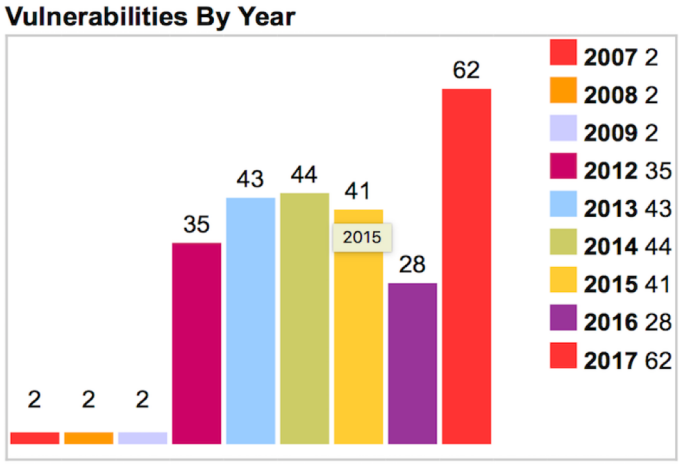


MirageOS running on ESP32 boards

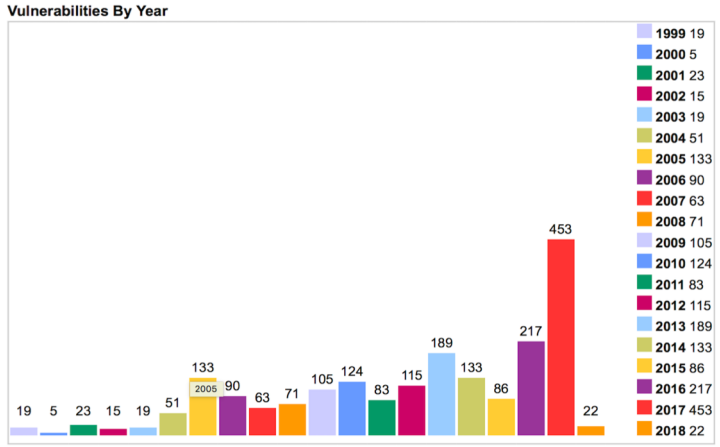


Motivation

- Minimal SW stack and interfaces are perfect for security
- But ?
- The system code can be malicious.
  - Traditional OSs and Hypervisors are large, complex, and buggy
  - This is more common than you think!



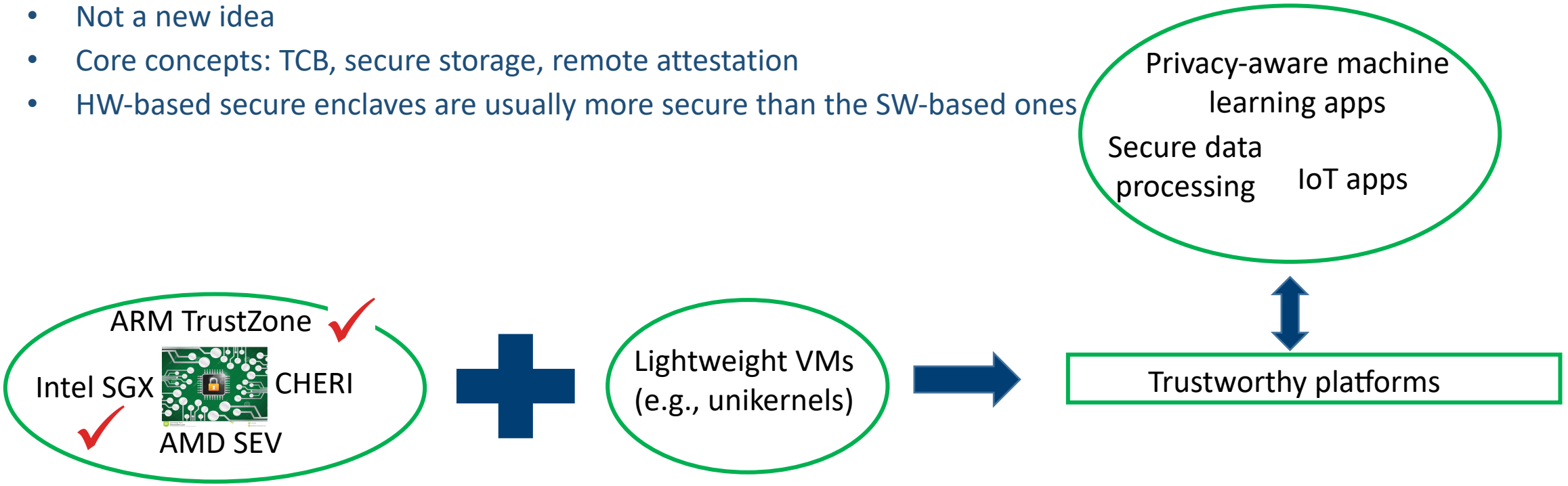
XEN vulnerabilities



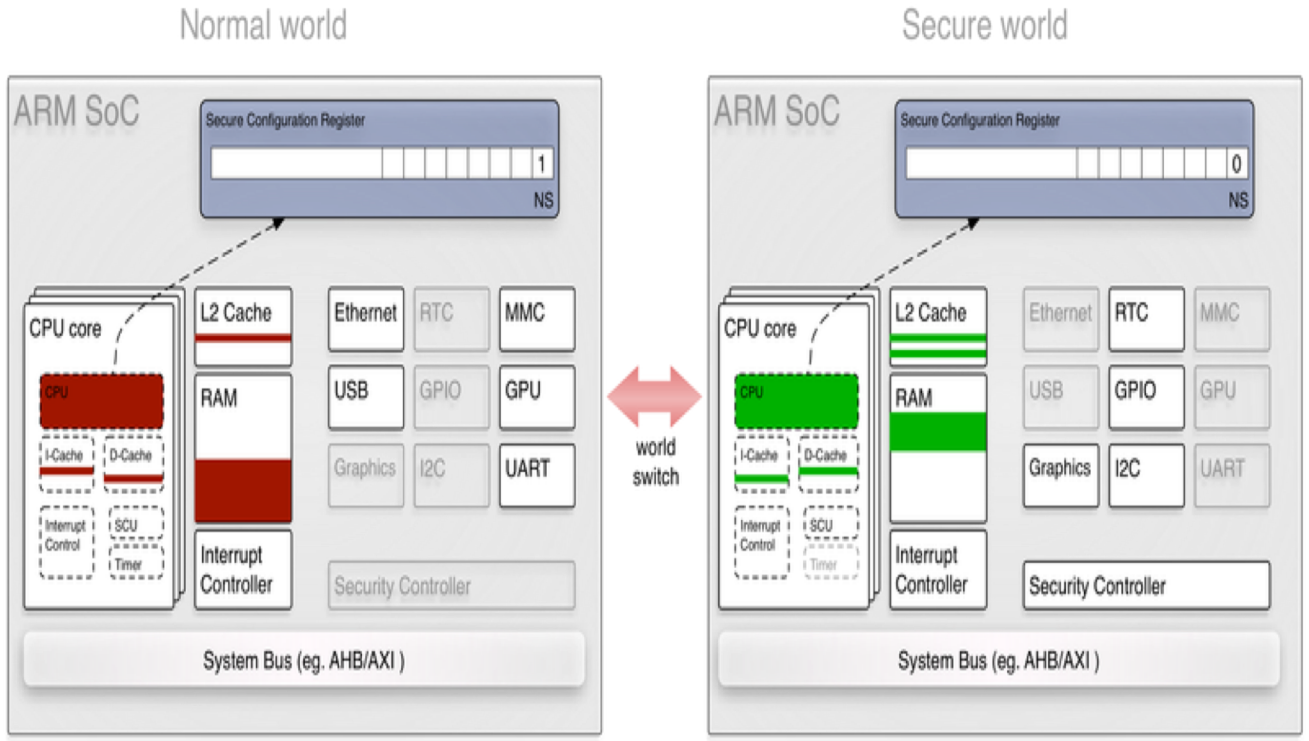
Linux kernel vulnerabilities

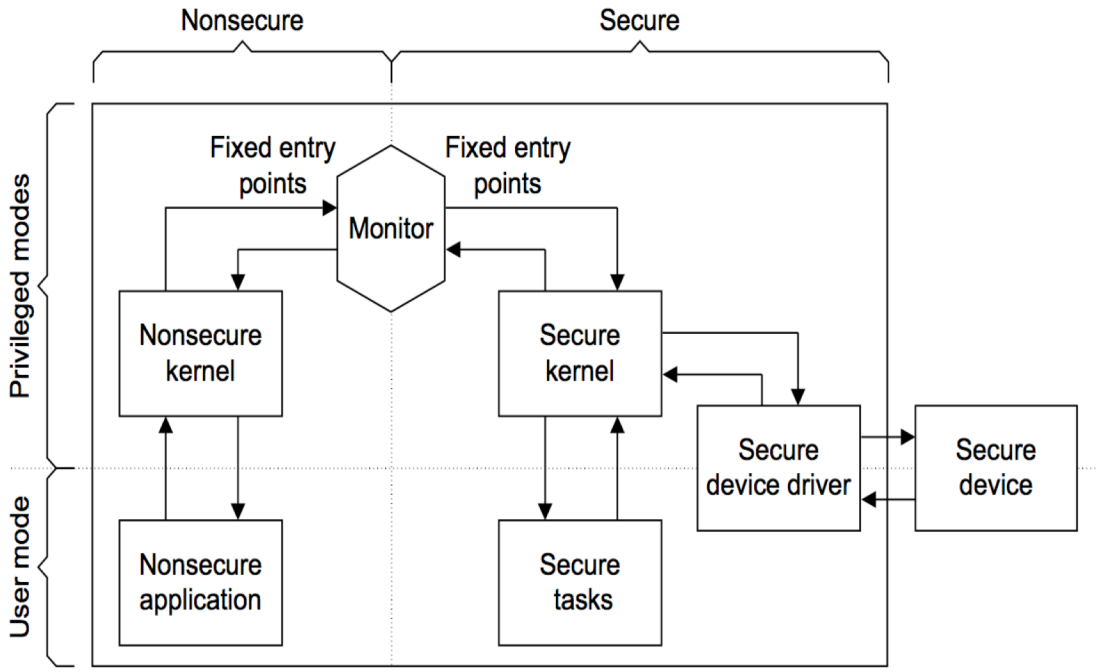
Motivation

- Trusted execution environments
  - Not a new idea
  - Core concepts: TCB, secure storage, remote attestation
  - HW-based secure enclaves are usually more secure than the SW-based ones



ARM TrustZone

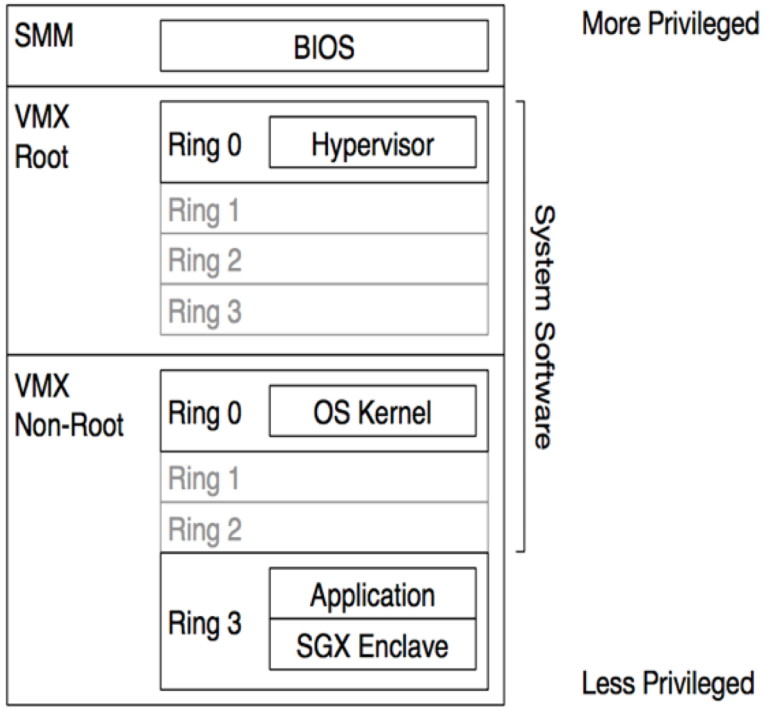




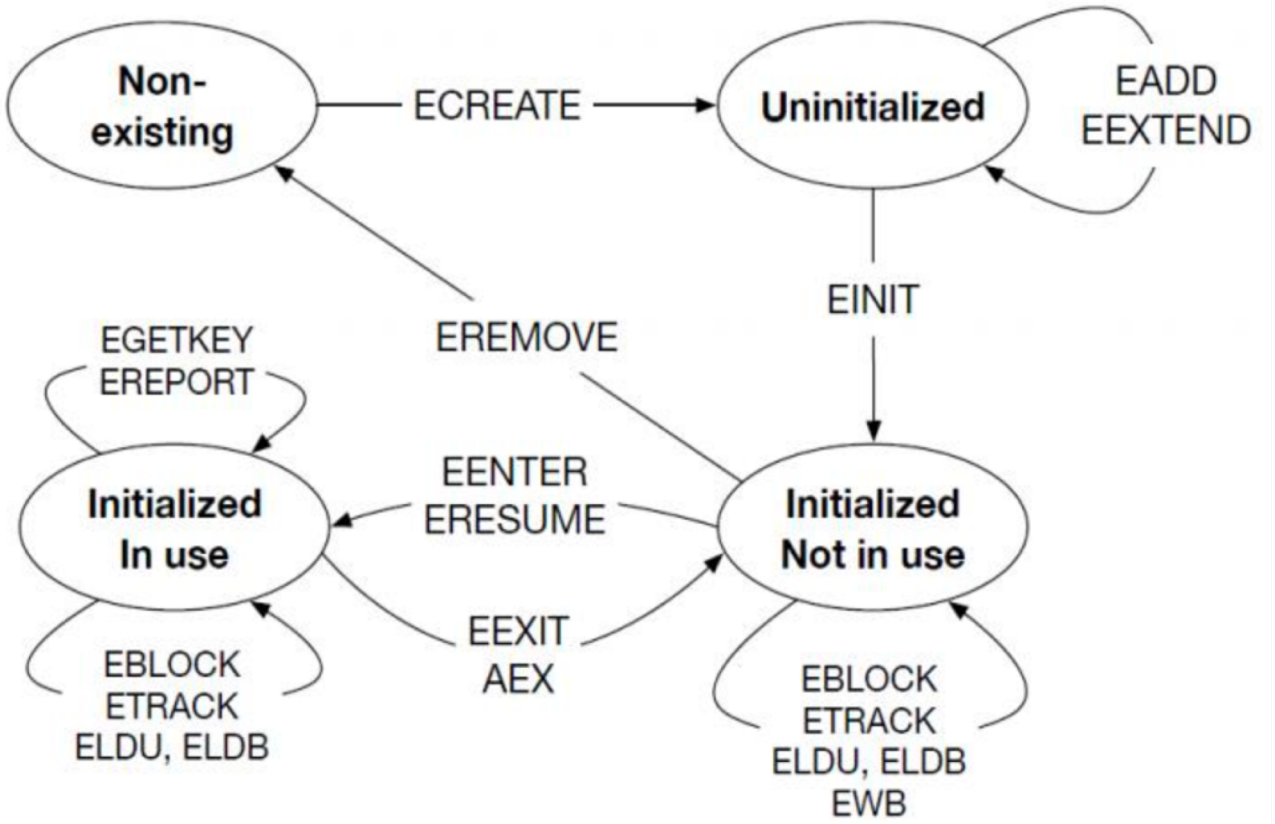
Intel SGX

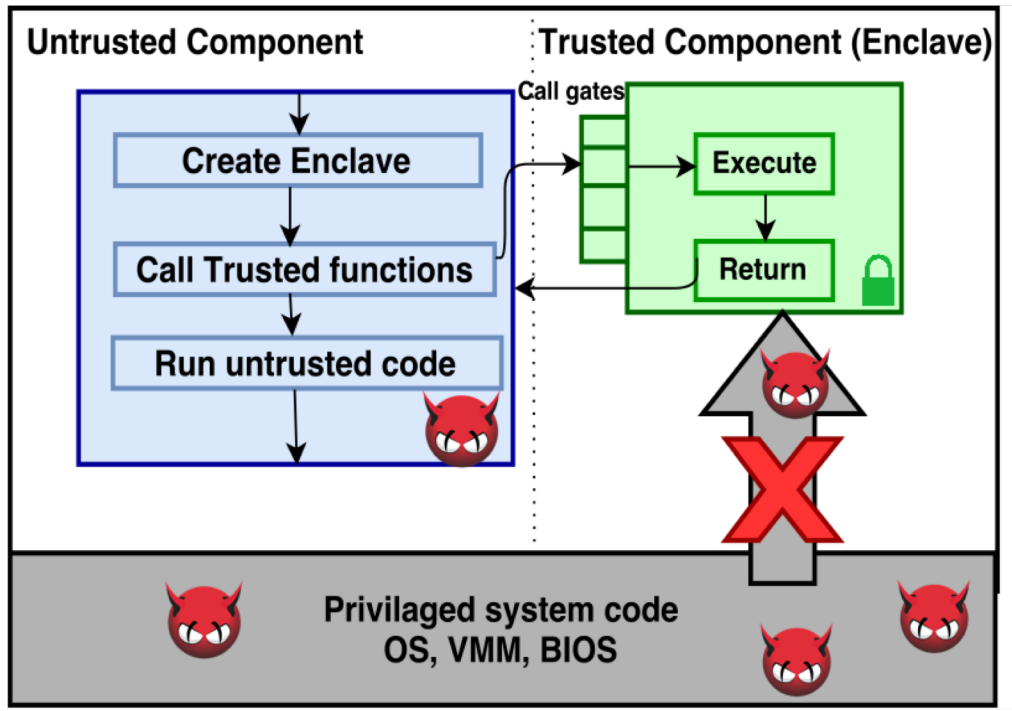
- SGX enclave is a protected area in the application's address space
- Protects the integrity and confidentiality of data/code from the system code
- Enclaves content is stored in the *Enclave Page Cache* (EPC)
  - EPC is a subset of Processor Reserved Memory (PRM)

- Avoids access from untrusted privileged code to enclaves code/data
  - Using access control mechanisms built into the processor
  - Memory encryption engine
- The host OS/Hyp handles page faults and resource managements
- No syscall and IO inside enclave



Intel SGX


Non-existing
ECREATE
Uninitialized
EADD
EEXTEND
EINIT
EREMOVE
EGETKEY
EREPORT
Initialized In use
EENTER
ERESUME
Initialized Not in use
EEXIT
AEX
EBLOCK
ETRACK
ELDU, ELDB
EBLOCK
ETRACK
ELDU, ELDB
EWB


Untrusted Component
Trusted Component (Enclave)
Call gates
Create Enclave
Call Trusted functions
Run untrusted code
Execute
Return
Privilaged system code
OS, VMM, BIOS

No technology is perfect

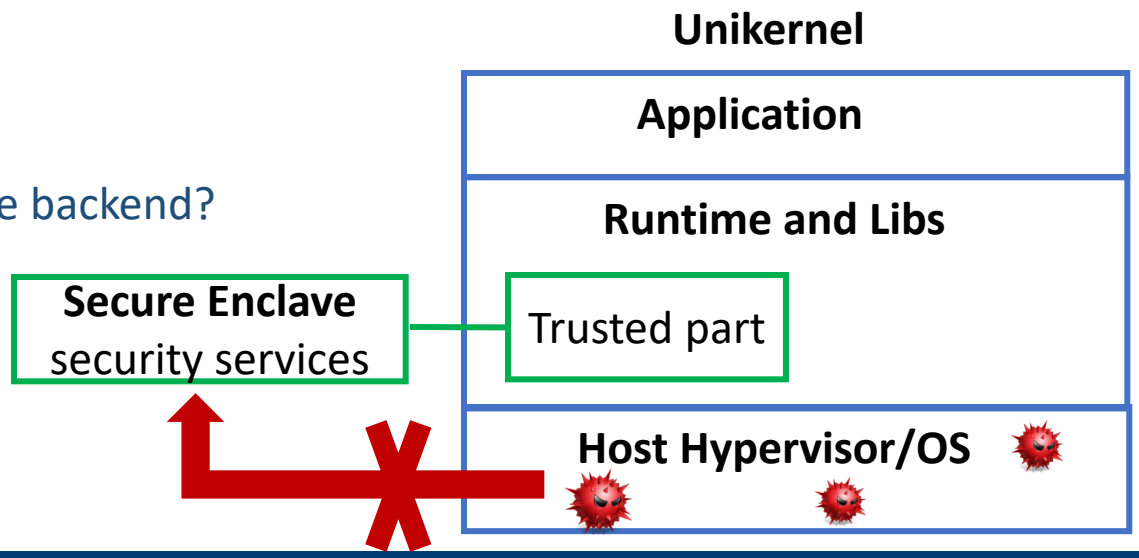
- Every thing is great so far, except secure enclaves **are not really secure**!

  - **SGX:**
    - Cache or page-fault channel attacks [Controlled-Channel Attacks[1], CacheZoom[2], Branch Shadowing[5])
    - Spectre attack [8,9]

  - **TZ:**
    - Channel attacks [TruSpy[15], ARMageddon[10]]
    - Limited public research
- And other issues: performance, scalability, resource limitation
- Lots of things to consider when designing a system using them

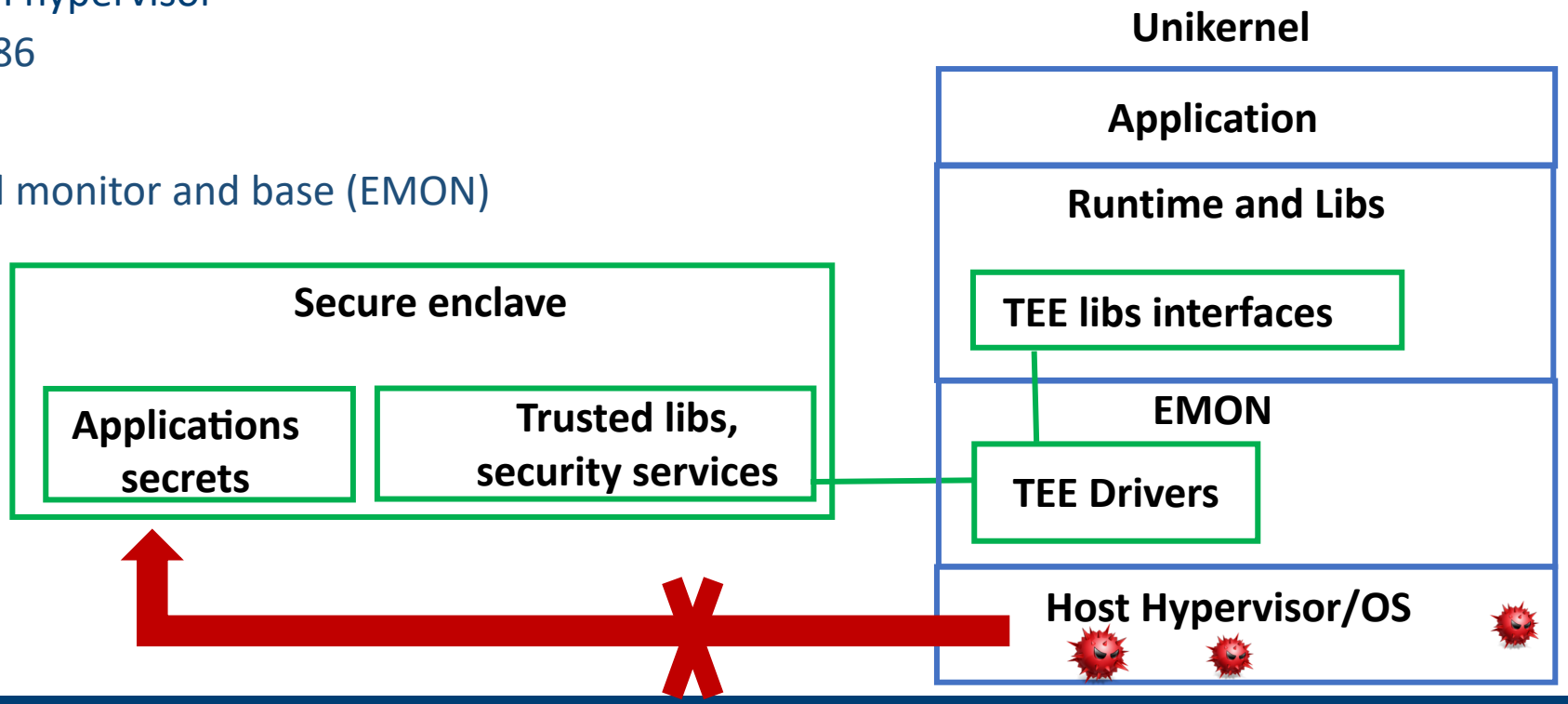
Trustworthy enclaves for unikernels

- The project is in early stage!
- High-level idea: provide minimal security services for different type of unikernel applications
  - The approach is different from Haven[11] or SCONE[12], do not want to isolate whole apps
  - Trusted libraries provide security services for unikernels
  - Security services really depends on applications needs
    - Secure storage/ secure file system
    - Secure NFV/middleboxes [lightbox, S-NFV]
    - Integrity monitoring and Attestation
  - Easy application deployment with different secure enclave backend?



Trustworthy enclaves for unikernels

- unikernel monitors:
  - Provides minimal hypervisor/emulations interfaces (e.g., ukvm)
  - Consider as lightweight Type-II hypervisor
  - ukvm works on ARM64 and X86
- Project current status:
  - Basic enclave-aware unikernel monitor and base (EMON)
    - TrustZone (on rpi3) , SGX



Possible usecases

- **Secure filesystem:**
  - ➢ Need to explore different designs:
  - A simple in-memory file system inside an enclave
  - having data blocks inside the enclave
  - Keep data blocks encrypted outside the enclave and manage keys inside the enclave
- **Secure edge/cloud data analytics**
  - Need an attestation mechanism for heterogeneous (SGX-TZ) enclaves
- **Secure NFV/middleboxes**
- (TZ specific) **Secure user interface/ device drivers**
- (TZ specific) **Integrity monitoring of unikernels at runtime**
- **Secure intermediate control units across the communication between IoT and Cloud.**

Challenges

- Secure enclaves limitations
- Heterogynous enclaves
- Real-time requirements, performance
- Sensitive multimedia streams and sensors data
- Shared secure resources
- Scalability

Questions?

References

- **[1]** Yuanzhong Xu, Weidong Cui, Marcus Peinado. *Controlled-channel attacks: Deterministic side channels for untrusted operating systems*. IEEE S&P, 2015.
- **[2]** Moghimi, Ahmad, Gorka Irazoqui, and Thomas Eisenbarth. "CacheZoom: How SGX Amplifies The Power of Cache Attacks.
- **[3]** J. Van Bulck, N. Weichbrodt, R. Kapitza et al. *Telling Your Secrets without Page Faults: Stealthy Page Table-Based Attacks on Enclaved Execution.* Usenix Security 17.
- **[4]** Sangho Lee, Ming-Wei Shih, Prasun Gera, et al. *Inferring Fine-grained Control Flow Inside SGX Enclaves with Branch Shadowing*. Usenix Security 17.
- **[5]** Lee, Sangho, et al. "Inferring fine-grained control flow inside SGX enclaves with branch shadowing." *26th USENIX Security Symposium, USENIX Security*. 2017.
- **[6]** Schwarz, M., Weiser, S., Gruss, D., Maurice, C., Mangard, S: *Malware guard extension: Using SGX to conceal cache attacks*. DIMVA 2017
- **[7]** S.Chen,X.Zhang,M.K.Reiter,andY.Zhang.Detecting privileged side-channel attacks in shielded executionwithDe ́ja ́ Vu. In*ACMSymposiumon Information, Computer and Communications Secu- rity*, 2017.

References

- **[8]** Chen, Guoxing, et al. "SgxPectre Attacks: Leaking Enclave Secrets via Speculative Execution." *arXiv preprint arXiv:1802.09085* (2018).
- **[9]** https://github.com/lsds/spectre-attack-sgx
- **[10]** Lipp, Moritz, et al. "ARMageddon: Cache Attacks on Mobile Devices." *USENIX Security Symposium*. 2016.
- **[11]** Baumann, Andrew, Marcus Peinado, and Galen Hunt. "Shielding applications from an untrusted cloud with haven." *ACM Transactions on Computer Systems (TOCS)* 33.3 (2015): 8.
- **[12]** Arnautov, Sergei, et al. "SCONE: Secure Linux Containers with Intel SGX." *OSDI*. Vol. 16. 2016.
- **[13]** Guan, Le, et al. "TrustShadow: Secure execution of unmodified applications with ARM trustzone." *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2017.
- **[14]** Madhavapeddy, Anil, et al. "Unikernels: Library operating systems for the cloud." *Acm Sigplan Notices*. Vol. 48. No. 4. ACM, 2013.
- **[15]** Zhang, Ning, et al. "TruSpy: Cache Side-Channel Information Leakage from the Secure World on ARM Devices." *IACR Cryptology ePrint Archive* 2016 (2016): 980.