Cooking Secrets in Leaky Cauldrons

the promises and promises of confidential computing

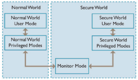
Nicolae Paladi

Secure co-processors 1999

Trusted Platform Module 1.2 2001

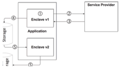
ARM TrustZone

2005



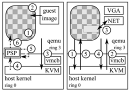
Intel SGX

2013



Trusted Platform Module 2.0 2014

AMD memory encryption 2016



TIME

TEE Cost/Processing Capability

SecureBlue++ 2011

TyTan 2015

Sanctum 2016

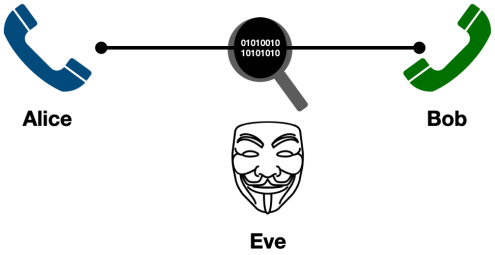
Graviton 2018

Software Guard Extensions

- Introduced in 2013;
- ISA extentions to set up **enclaves** that isolate execution environments from other applications, the operating systems kernel and the hypervisor;
- Enclaves can run **arbitrary code** on commodity hardware (different from earlier hardware co-processors).
- **Attestation**: enclave can prove to a remote entity that it was instantiated correctly.

Adversary Model

- Trusted CPU
- Trusted code running in enclaves
- Untrusted host
- Adversary controls OS



(Some) Applications

- *Trust anchors in software defined networks* [PKE2018]
  - Shielding **credentials and crypto libraries** for software network components
- *Protecting OpenFlow with SGX* [MPA2019]
  - Open vSwitch decomposed to port **routing tables to SGX enclaves**
- *SDN access control for the masses* [PG2019]
  - **Running a security monitor** for access control in Software Defined Networks
- *Privacy for vulnerability recommendations* [KP2019]
  - Recommending and prioritizing software patches **without revealing the structure of the enterprise software stack**

Leaky Cauldrons

Controlled-channel attacks

RI.
SE

Main idea

Make use of the (near-total) control the OS has over the platform

- Monitoring or introducing page-faults
- Configuring the APIC timers, issuing interrupts and tracking page-table entries
- Timing carefully synchronized interrupts while the enclave is running
- Resetting the *accessed* flag in the translation lookaside buffer

Cache attacks

Main idea

- Exploiting the cache-hierarchy system
- Caching memory loads from DRAM leaves effects on the system state measurable from outside the protected application

- SGX enclaves are vulnerable to same cache attacks against secret information as any other software application :(
- Many techniques for side-channel extraction: *Flush+Reload*, *Prime+Probe*, *Evict+Time*, *Evict+Reload*, *Flash+Flush*, etc.

Speculative execution attacks

RI.
SE

Main idea

Speculatively executed CPU instructions cause measurable changes in the CPU state

- Initially thought to only affect "regular" execution
- SgxPectre attack and used it to extract the secret seal keys and attestation keys from Intel signed quoting enclaves

Rogue Data Cache Loads

RI.
SE

Main idea

Leverage speculative out-of-order execution but use a race condition where results of unauthorized memory accesses remain transiently available

- CPU issues a fault and rolls-back the results of out-of-order execution instructions **too late**.
- Affects the CPU cache and allows the memory access to be inferred.

Microarchitectural Data Sampling (MDS)

Main idea

Leverage leakage of information from implementation-specific undocumented intermediary buffers of the target micro architecture.

- Bypasses most common security boundaries: JavaScript sandboxes, processes, kernels, VMs and SGX enclaves.
- Works despite mitigations to branch prediction attacks and speculative execution attacks
- Attack enclaves cross-core (across execution units) by exploiting an **undocumented** *staging buffer* shared between all cores.
- CacheOut, CrossTalk, SGX-Axe published recently

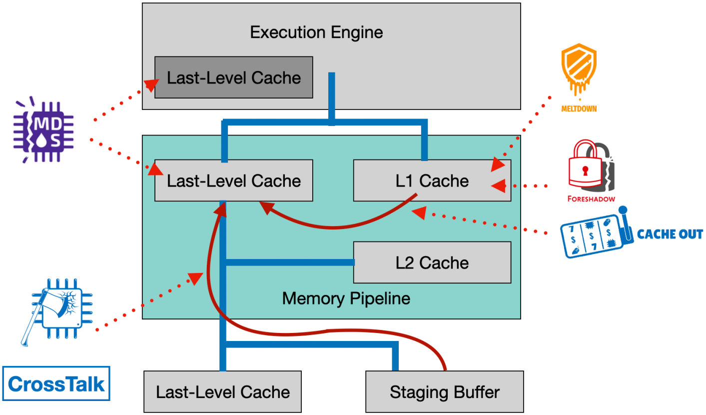
Software-based Fault Injection Attacks

Main idea

Leverage privileged interfaces for dynamic voltage scaling on the x86 CPU to corrupt enclave computations.

- Briefly decreasing the CPU voltage during computation allows a privileged adversary to inject faults into protected enclave computations

Overview

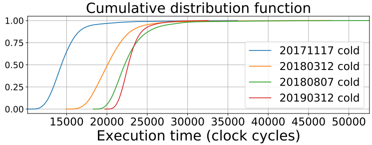


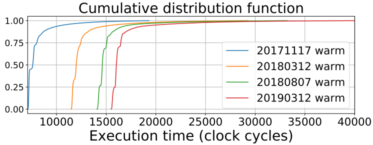
Execution Engine
Last-Level Cache
Last-Level Cache
L1 Cache
L2 Cache
Memory Pipeline
Last-Level Cache
Staging Buffer
MDS
CrossTalk
MELTDOWN
FORESHADOW
CACHE OUT

Countermeasures

Microcode Patch

- Most complex CPU instructions are implemented in microcode
- Effective but costly (e.g. disable hyperthreading)
- Mitigated using **microcode patches**:
  - many Speculative Execution Attacks
  - Rogue Data Cache Loads
  - MDS Attacks
  - Software Fault Injection Attacks





System Redesign

- Redesigning or removing implementation issues in **supporting** systems
- Examples of supporting systems: *Launcher Enclave* (LE), *Provisioning Enclave* (PE), *Quoting Enclave* (QE).
- May take a long time and require waiting for the next platform generation.

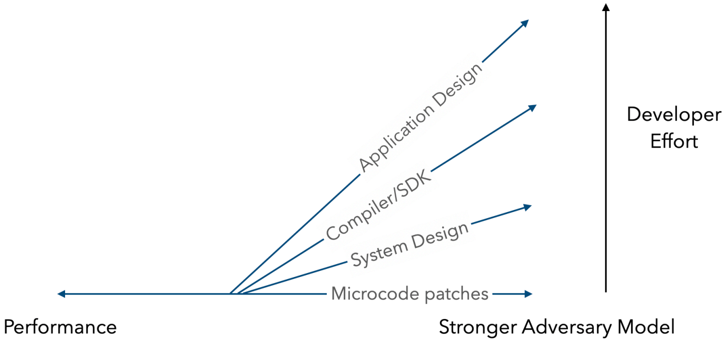
Compiler or SDK

- **Better strategy**: leave the responsibility to someone else (developers!)
- Hoping for a general solution: **a better compiler or SDK**
- Might come with various **performance impacts** and **other drawbacks**
- In many cases this makes the exploitation *more difficult* but not impossible

Application Design

- Leave the responsibility to someone else (developers!)
  - Wrapping secret data & code into **TSX transactions**
  - Using **Oblivious RAM** to hide memory access patterns
- Costly mitigations...
- Can fix many (most?) **Side Channel Attacks**

Overivew



Application Design
Compiler/SDK
System Design
Microcode patches
Developer Effort
Performance
Stronger Adversary Model

References I

[BOMAGP2019] J. Van Bulck, D. Oswald, E. Marin, A. Aldoseri, F. D. Garcia, and F. Piessens,
'A Tale of Two Worlds'.
Proc. 2019*ACM SIGSAC Conf. Comput. Commun. Secur. - CCS 19*

[PKE2018] Paladi, Nicolae, Linus Karlsson, and Khalid Elbashir.
'Trust anchors in software defined networks.'.
European Symposium on Research in Computer Security (ESORICS), Springer, Cham, 2018.

[KP2019] Karlsson, Linus, and Nicolae Paladi.
'Privacy-enabled Recommendations for Software Vulnerabilities.'.
2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing (DASC). IEEE, 2019.

References II

[PG2019] Paladi, Nicolae, and Christian Gehrmann.
'SDN access control for the masses.'.
Computers and Security 80 (2019): 155-172.

[NNB2019] Nilsson, Alexander, Pegah Nikbakht Bideh, and Joakim Brorsson.
'A survey of published attacks on intel SGX.'.
arXiv preprint arXiv:2006.13598 (2020)

[GY2021] Genkin, Daniel, and Yuval Yarom..
'Whack-a-Meltdown: Microarchitectural Security Games [Systems Attacks and Defenses].'.
IEEE Security Privacy 19.1 (2021): 95-98