Microarchitecture security

PHISIC 2019

Mathieu Escouteloup (INRIA Rennes)

Advisors: Ronan Lashermes (INRIA Rennes), Jean-Louis Lanet (INRIA Rennes), Jacques Fournier (CEA)

mathieu.escouteloup@inria.fr

October 15th, 2019

Spectre and Meltdown impact

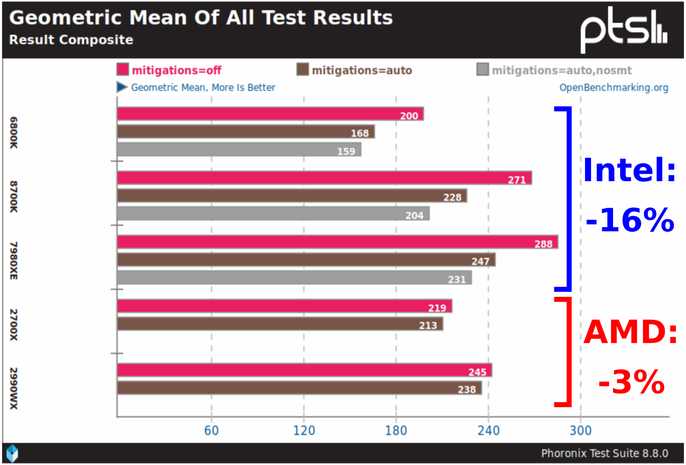


Figure: Mitigations impact on performance on Intel CPUs[1]

[1] From "The Performance Impact Of MDS / Zombieload Plus The Overall Cost Now Of Spectre/Meltdown/L1TF/MDS" on www.phoronix.com

Table of contents

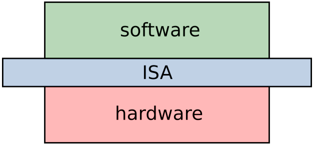
1. Attacks principles
2. Countermeasure methodology
3. Microarchitecture, ISA and security
4. Conclusion

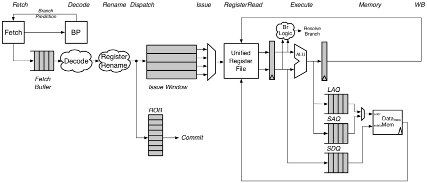
Table of contents

1. Attacks principles
2. Countermeasure methodology
3. Microarchitecture, ISA and security
4. Conclusion

Overview

Figure: RISC-V BOOM core microarchitecture[2]

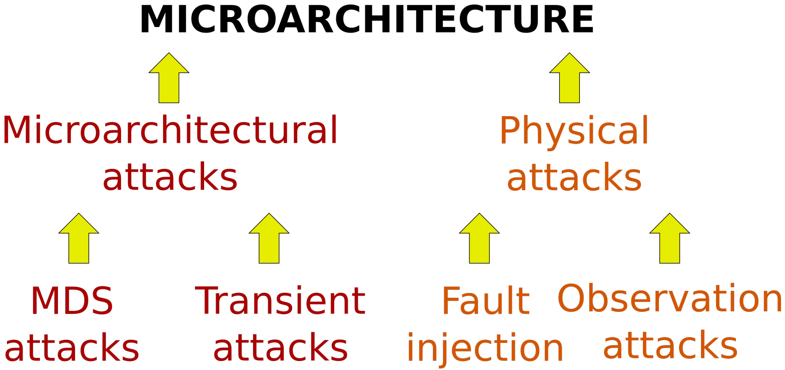




- Focus on attacks affecting the microarchitecture.
- Different possible origins.

[2] from https://docs.boom-core.org/en/latest/index.html

Attacks hierarchy



MICROARCHITECTURE

Microarchitectural attacks

Physical attacks

MDS attacks

Transient attacks

Fault injection

Observation attacks

MDS: Microarchitectural Data Sampling

Microarchitectural attacks

1.x **Transient attacks:** exploit instructions executed but not comitted.

1.1 **Spectre-class:** exploit speculation mechanisms.

1.2 **Meltdown-class:** transfer data from a forbidden instruction.

2.x **MDS-class:** exploit data leakage from shared resources.

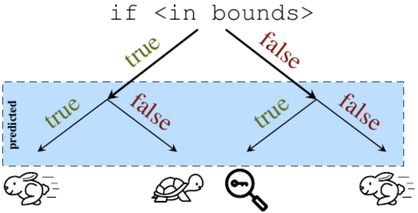
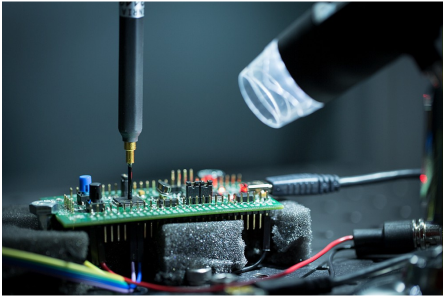


Figure: Speculative execution assumption [3]

[3] From "Spectre Attacks: Exploiting Speculative Execution", P. Kocher et al., S&P'19

Physical fault injection



- Principle: disturb chip environment to modify signal values
- Exploitation: modify data, executed operations ...

Attacks principles

ISA: a broken interface

- Transient attacks: execution sequencing not respected.
- Fault injection: altered instructions.
- Observation attacks: instructions leak informations.

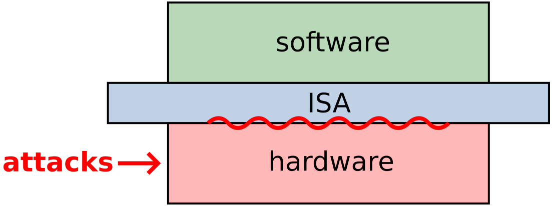


Table of contents

1. Attacks principles
2. Countermeasure methodology
3. Microarchitecture, ISA and security
4. Conclusion

Security cycle: the reactive model

1 A weakness is discovered.

2 It is studied and solutions are considered.

3 A patch is applied.

... Finally: mitigations stacking.

From a specific countermeasure ...

Retpoline

- Spectre-BTB[a] (variant 2) mitigation used in Windows 10.
- Designed as a compilation pass.
- Replace indirect jumps by a return sequence.
- Goal: do not use the BTB ...

[a] BTB: Branch Target Buffer

jmp *%r11	call set_up_target; **(1)** **capture_spec: (3b)** pause; jmp capture_spec; **set_up_target:** mov (%rsp), %r11; **(2)** ret; **(3a)**

… to a global solution.

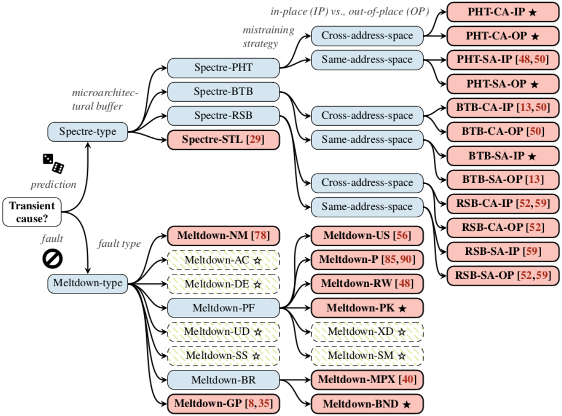


Figure: Transient attacks classification[4]

[4] From “A Systematic Evaluation of Transient Execution Attacks and Defenses”, C. Canella et al., USENIX Security’19

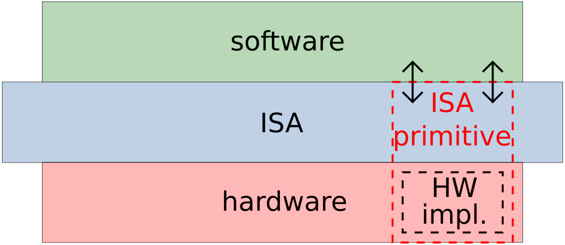
Table of contents

1. Attacks principles
2. Countermeasure methodology
3. Microarchitecture, ISA and security
4. Conclusion

ISA and security

ISA: a main role

- **Define needed security guarantees.**
- **Constrain microarchitecture design for security.**
- **Make some primitives available for the software.**



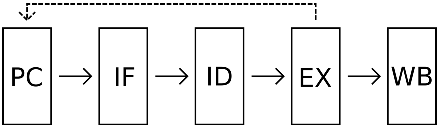
Current work: hardware contexts

Principles

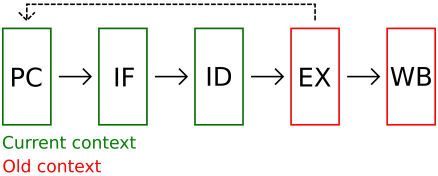
- Introduce a notion of execution context at the ISA-level.
- Use a context identifier to define a security domain.
- A context change is also a security domain change.
- Application: a tool to know when data can be shared.

Hardware contexts: possible implementations

A simple microarchitecture representation



With hardware contexts



Hardware contexts: an application example

Partitioned BTB

- Each value is linked with a context.
- Here, mitigate Spectre-BTB.
- Extensible to other hardware mechanisms (speculation, cache memories ...).

current address 0	target address X	**context 0**
current address 5	target address N	**context 0**
current address 0	target address Z	**context 2**
current address 3	target address Y	**context 1**

Table of contents

1. Attacks principles
2. Countermeasure methodology
3. Microarchitecture, ISA and security
4. Conclusion

Conclusion

Global view

- New weaknesses regularly discovered on modern microarchitectures.
- Complexity is still increasing: 2.186.259 words in x86 specification.
- Integrate security assumptions from the beginning.

Our work

- Define security guarantees at the ISA-level.
- Evaluate hardware contexts with a real implementation.

Conclusion

Other possible workpaths

- **CFI[a]: why not ban indirect jumps ?**
- **Define and constraint hardware features: RNG[b].**
- **Instructions with constant time constraints.**
- **Specific GPRs[c] only usable by secure instructions.**

[a] CFI: Control-Flow Integrity

[b] RNG: Random Number Generator

[c] GPR: Global Purpose Register

Microarchitecture security
PHISIC 2019

Mathieu Escouteloup (INRIA Rennes)

Advisors: Ronan Lashermes (INRIA Rennes), Jean-Louis Lanet (INRIA Rennes), Jacques Fournier (CEA)

mathieu.escouteloup@inria.fr

October 15th, 2019