Introduction to software-based microarchitectural side-channel attacks

Abc Xyz
@dura_lex

DCG#7812
2018

Agenda

Introduction

Theory

Basic attacks

Software-based Microarchitectural Fault Attacks

Meltdown & Spectre

Summary

Agenda

Introduction

Side-channel attacks

Microarchitectural attacks

Agenda

Introduction

Side-channel attacks

Side-channel attacks

Example of target for side-channel attack

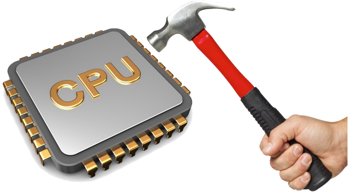
Agenda

Introduction

- Microarchitectural attacks

Microarchitectural attacks

```
code1a:
  mov (X), %eax
  mov (Y), %ebx
  clflush (X)
  clflush (Y)
  jmp code1a
```



The DRAM cells get permanently damaged if hammered for a long time

Agenda

Theory

- CPU
- Cache
- DRAM

Agenda

Theory

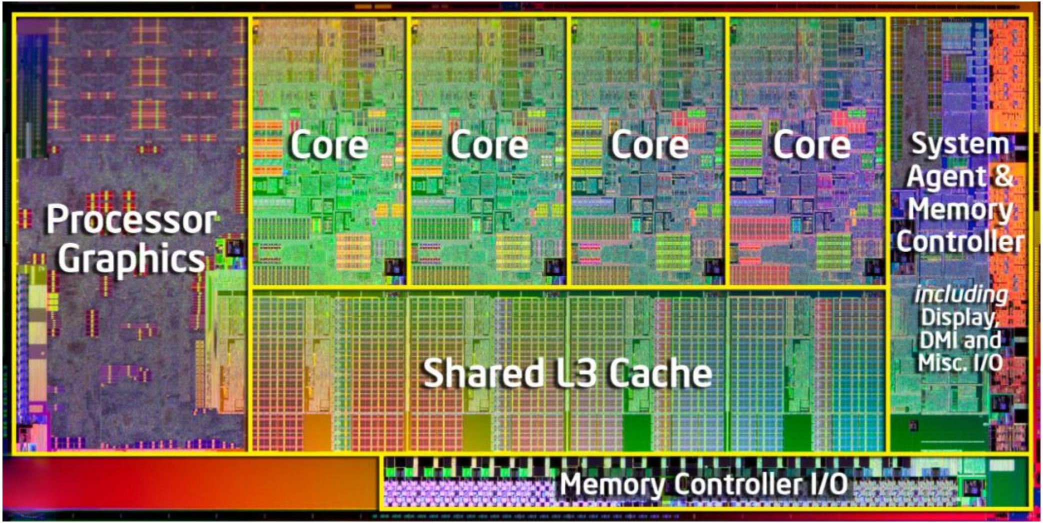
CPU

Pipelining

Branch Prediction and Speculation

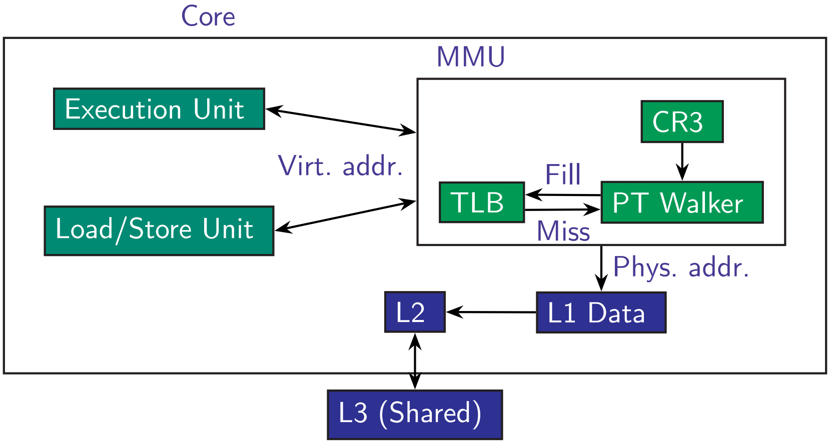
Multicore

CPU



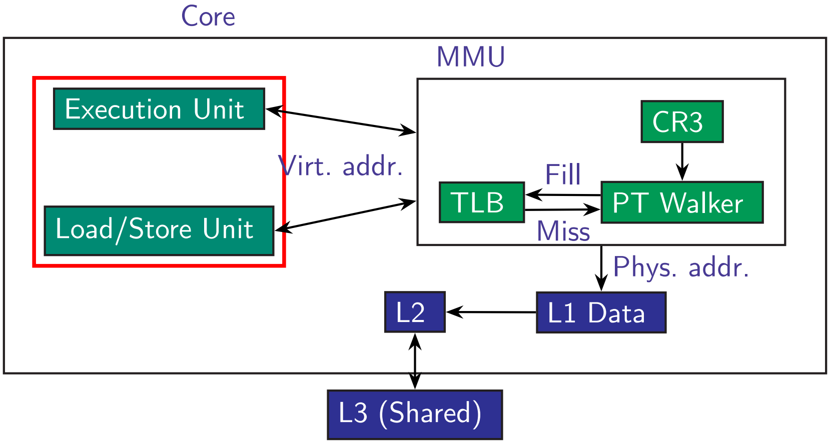
Architecture of multicore CPU

CPU



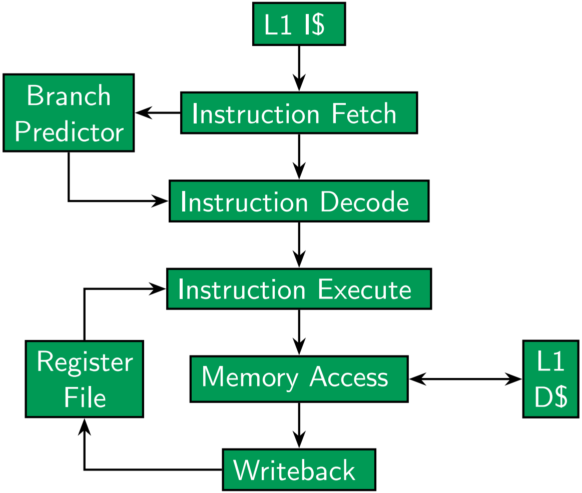
Abstract architecture of core and memory organization

CPU



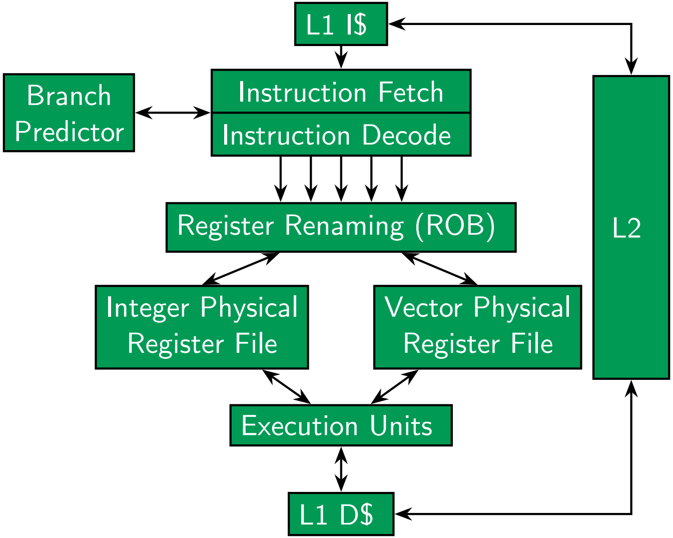
Abstract architecture of core and memory organization

Pipelining. In-Order



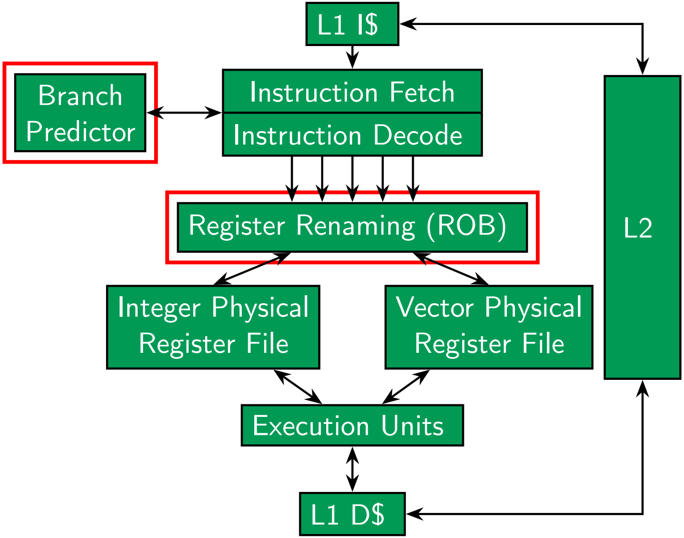
Elements of a modern in-order core

Pipelining. Out-of-Order



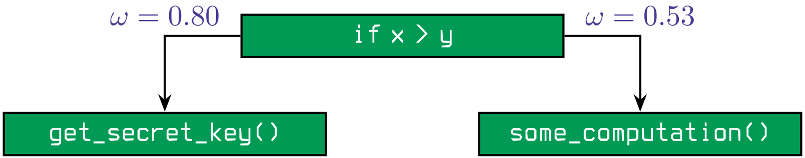
Elements of a modern out-of-order core

Pipelining. Out-of-Order



Elements of a modern out-of-order core

Branch Prediction and Speculation



$\omega = 0.80$

`if x > y`

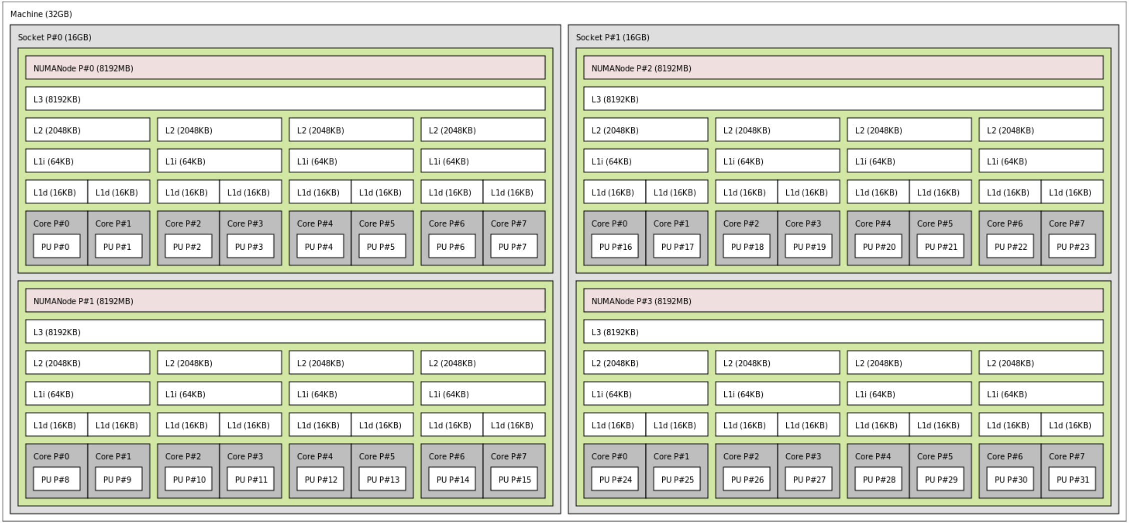
$\omega = 0.53$

`get_secret_key()`

`some_computation()`

`get_secret_key()` can be executed speculatively

Multicore



Machine (32GB)

Socket P#0 (16GB)

NUMANode P#0 (8192MB)

L3 (8192KB)

L2 (2048KB) L2 (2048KB) L2 (2048KB) L2 (2048KB)

L1i (64KB) L1i (64KB) L1i (64KB) L1i (64KB)

L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB)

Core P#0 PU P#0 | Core P#1 PU P#1 | Core P#2 PU P#2 | Core P#3 PU P#3 | Core P#4 PU P#4 | Core P#5 PU P#5 | Core P#6 PU P#6 | Core P#7 PU P#7

NUMANode P#1 (8192MB)

L3 (8192KB)

L2 (2048KB) L2 (2048KB) L2 (2048KB) L2 (2048KB)

L1i (64KB) L1i (64KB) L1i (64KB) L1i (64KB)

L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB)

Core P#0 PU P#8 | Core P#1 PU P#9 | Core P#2 PU P#10 | Core P#3 PU P#11 | Core P#4 PU P#12 | Core P#5 PU P#13 | Core P#6 PU P#14 | Core P#7 PU P#15

Socket P#1 (16GB)

NUMANode P#2 (8192MB)

L3 (8192KB)

L2 (2048KB) L2 (2048KB) L2 (2048KB) L2 (2048KB)

L1i (64KB) L1i (64KB) L1i (64KB) L1i (64KB)

L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB)

Core P#0 PU P#16 | Core P#1 PU P#17 | Core P#2 PU P#18 | Core P#3 PU P#19 | Core P#4 PU P#20 | Core P#5 PU P#21 | Core P#6 PU P#22 | Core P#7 PU P#23

NUMANode P#3 (8192MB)

L3 (8192KB)

L2 (2048KB) L2 (2048KB) L2 (2048KB) L2 (2048KB)

L1i (64KB) L1i (64KB) L1i (64KB) L1i (64KB)

L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB) L1d (16KB)

Core P#0 PU P#24 | Core P#1 PU P#25 | Core P#2 PU P#26 | Core P#3 PU P#27 | Core P#4 PU P#28 | Core P#5 PU P#29 | Core P#6 PU P#30 | Core P#7 PU P#31

Architecture of multicore CPU AMD Bulldozer

Agenda

Theory

Cache

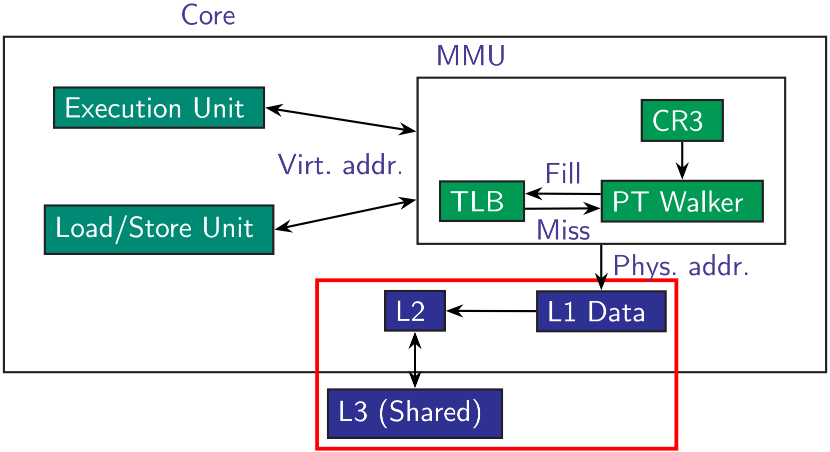
Types of cache

Two-way set associative cache

Cache replacement policies

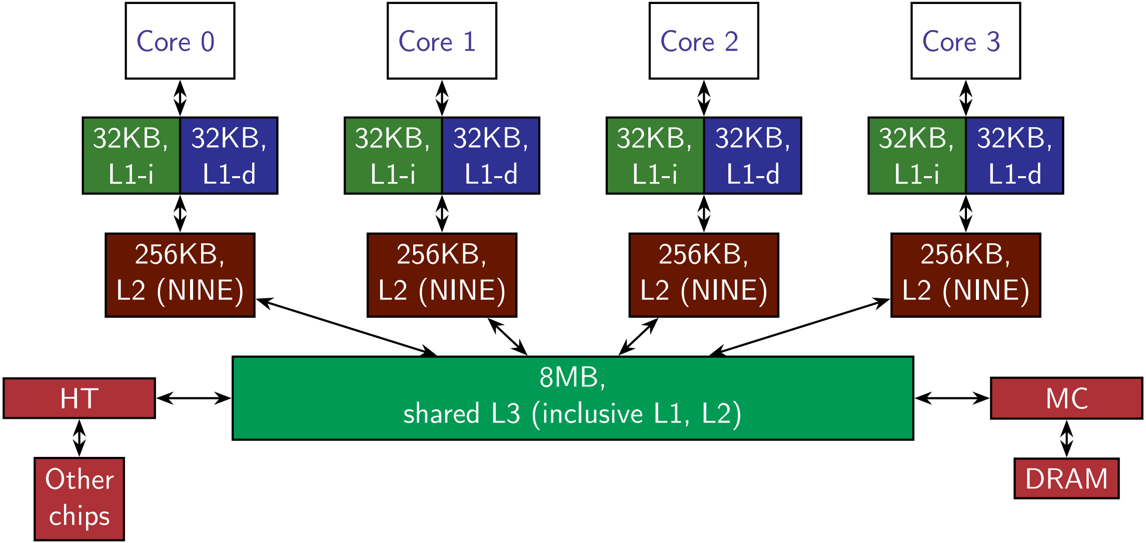
Addressing modes

Cache



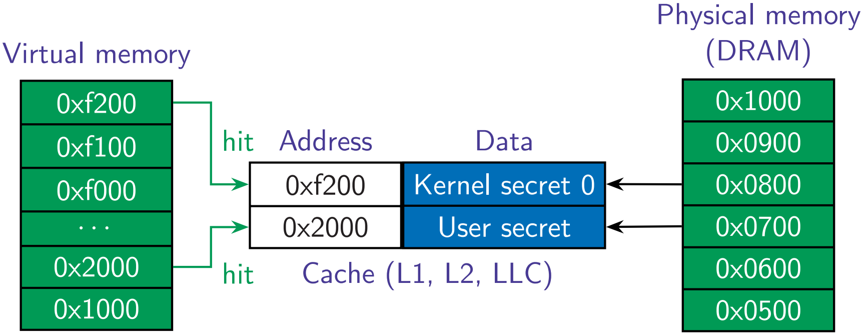
Abstract architecture of core and memory organization

Cache



The flow of data through a modern platform

Cache


Virtual memory

0xf200

0xf100

0xf000

· · ·

0x2000

0x1000

hit

hit

Address	Data
0xf200	Kernel secret 0
0x2000	User secret

Cache (L1, L2, LLC)

Physical memory (DRAM)

0x1000

0x0900

0x0800

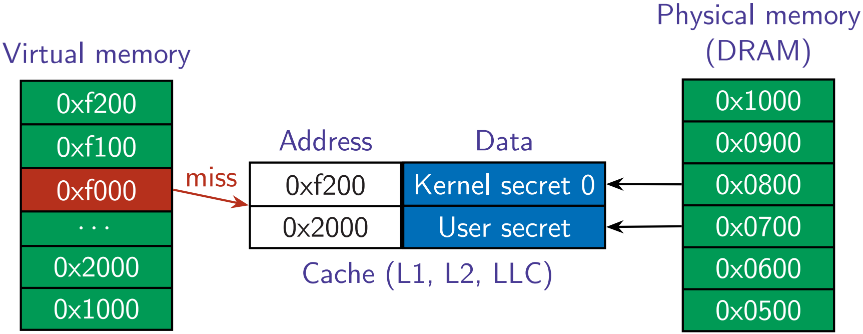
0x0700

0x0600

0x0500

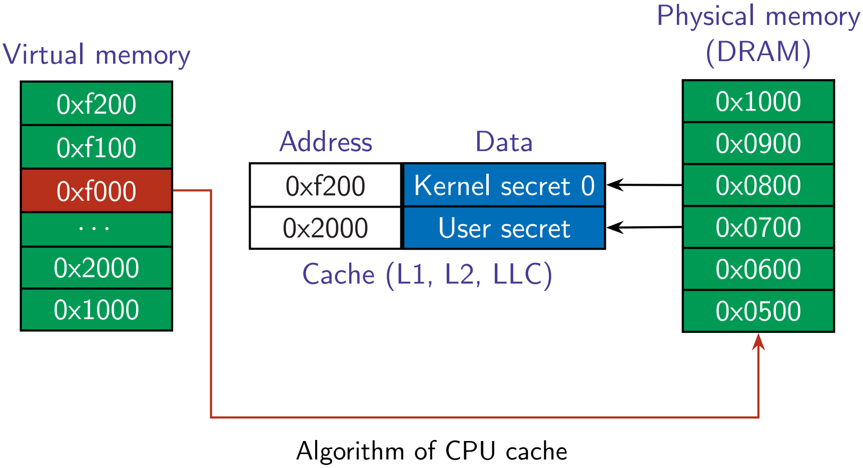
Algorithm of CPU cache

Cache

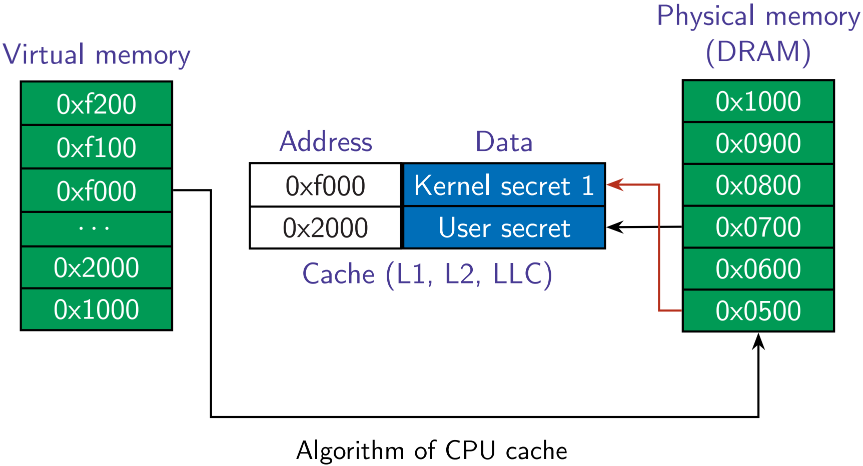


Algorithm of CPU cache

Cache


Virtual memory
0xf200
0xf100
0xf000
· · ·
0x2000
0x1000
Address
Data
0xf200
Kernel secret 0
0x2000
User secret
Cache (L1, L2, LLC)
Physical memory (DRAM)
0x1000
0x0900
0x0800
0x0700
0x0600
0x0500
Algorithm of CPU cache

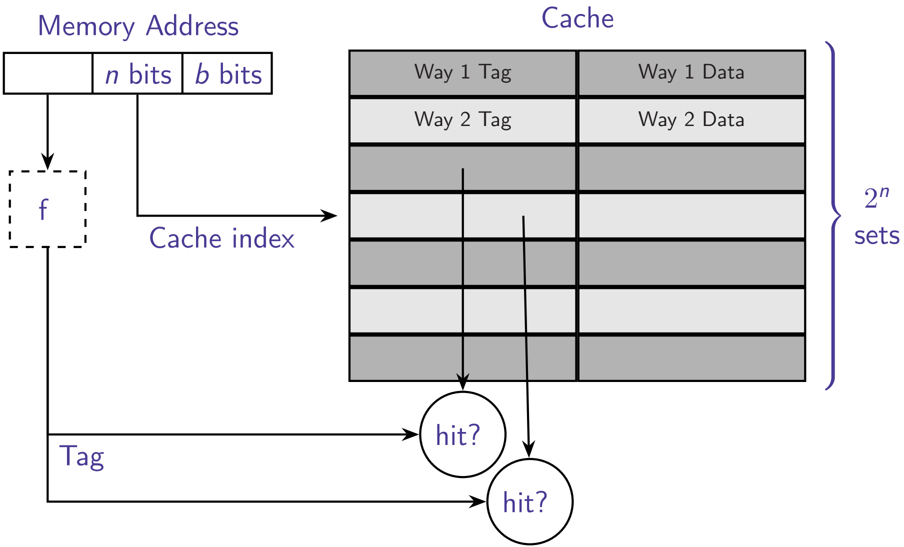
Cache


Virtual memory
0xf200
0xf100
0xf000
. . .
0x2000
0x1000
Address
Data
0xf000
Kernel secret 1
0x2000
User secret
Cache (L1, L2, LLC)
Physical memory
(DRAM)
0x1000
0x0900
0x0800
0x0700
0x0600
0x0500
Algorithm of CPU cache

Types of cache

- Direct-mapped cache
- Fully-associative cache
- 2/4/8/12-way set associative cache

Two-way set associative cache



Cache replacement policies

- FIFO
- LIFO
- least recently used, LRU
- time aware least recently used, TLRU
- most recently used, MRU
- pseudo-LRU, PLRU
- random replacement, RR
- segment LRU, SLRU
- least frequently used, LFU
- least frequent recently used, LFRU
- LFU with dynamic aging, LFUDA
- low inter–reference recency set, LIRS
- adaptive replacement cache, ARC
- clock with adaptive replacement, CAR
- multi queue, MQ
- and etc.

Addressing modes

- Virtually indexed, virtually tagged (VIVT)
- Physically indexed, virtually tagged (PIVT)
- Virtually indexed, physically tagged (VIPT)
- Physically indexed, physically tagged (PIPT)

Agenda

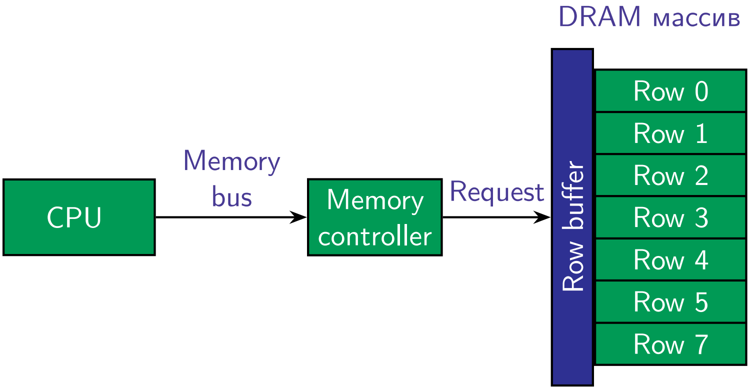
Theory

DRAM

How DRAM works

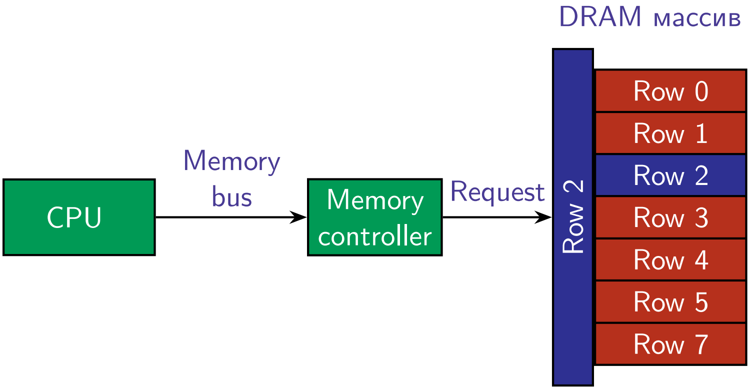
DRAM organization

How DRAM works



A very simple computer system, with a single DRAM array

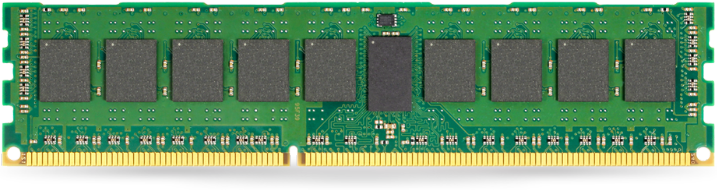
How DRAM works



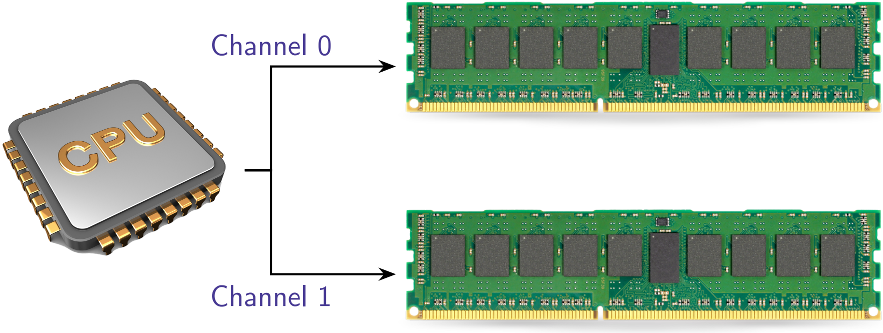
A very simple computer system, with a single DRAM array

DRAM organization

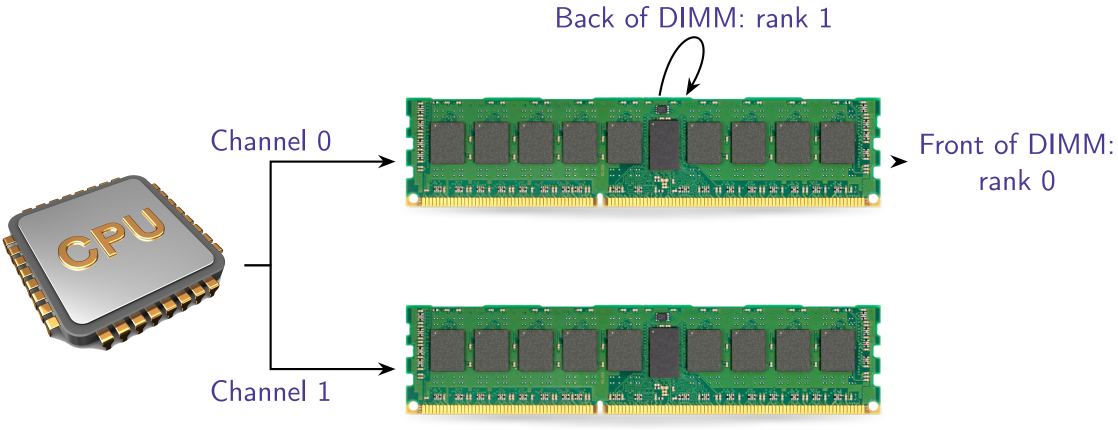
DIMM



DRAM organization



DRAM organization



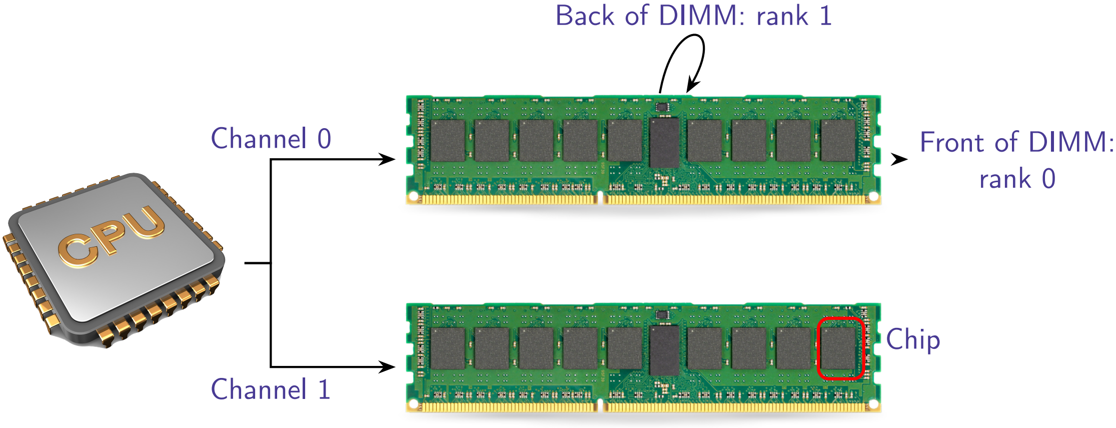
Back of DIMM: rank 1

Channel 0

Front of DIMM: rank 0

Channel 1

DRAM organization



Back of DIMM: rank 1

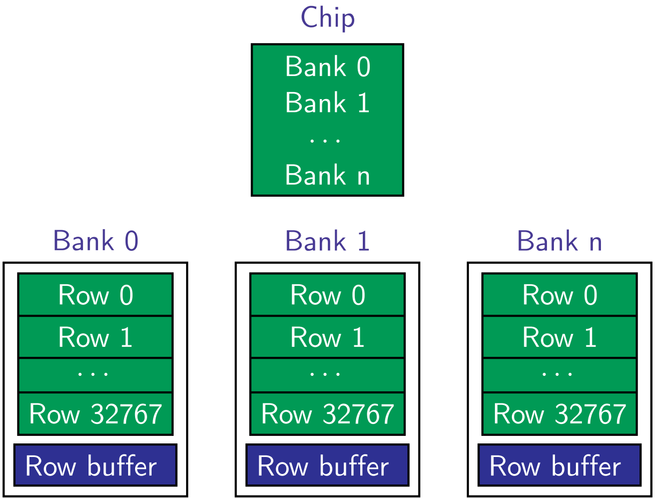
Channel 0

Front of DIMM: rank 0

Chip

Channel 1

DRAM organization



Chip

Bank 0
Bank 1
. . .
Bank n

Bank 0

Row 0
Row 1
. . .
Row 32767

Row buffer

Bank 1

Row 0
Row 1
. . .
Row 32767

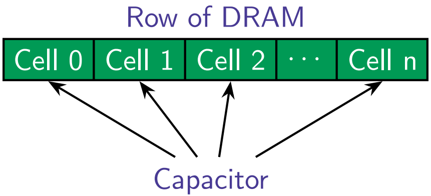
Row buffer

Bank n

Row 0
Row 1
. . .
Row 32767

Row buffer

DRAM organization



Row of DRAM

Cell 0	Cell 1	Cell 2	· · ·	Cell n

Capacitor

Agenda

Basic attacks

- Cache attacks
- Branch-prediction attacks
- TLB-based attacks
- Exception-based attacks
- DRAM-based attacks
- Covert channels

Agenda

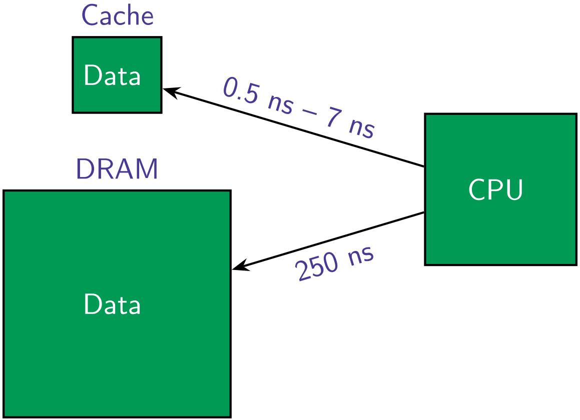
Basic attacks

Cache attacks

Flush + Reload

Cache attacks

Cache attacks



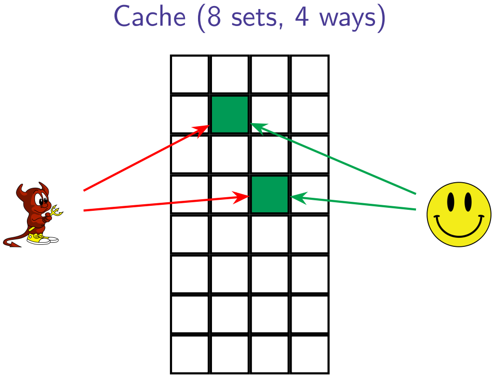
Timing attack — attack exploiting differences in the execution time of an algorithm

Flush + Reload

1. Map binary (e.g., shared object) into address space
2. Flush a cache line (code or data) from the cache
3. Schedule the victim's program
4. Check if corresponding cache line from step 2 has been loaded by the victim's program

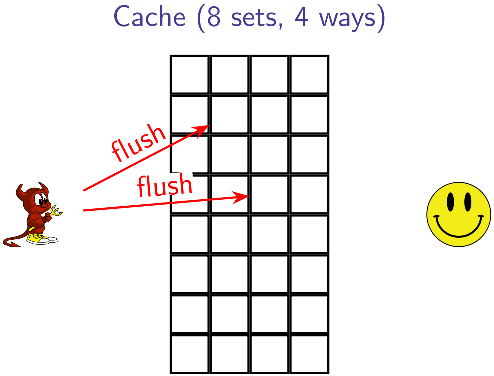
Flush + Reload

Map binary (e.g., shared object) into address space



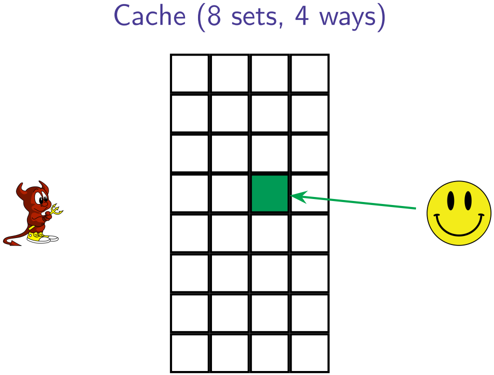
Flush + Reload

Flush a cache line (code or data) from the cache



Flush + Reload

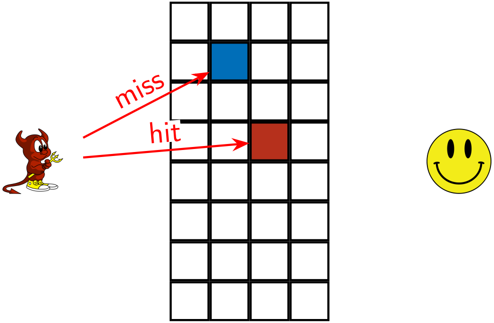
Schedule the victim's program



Flush + Reload

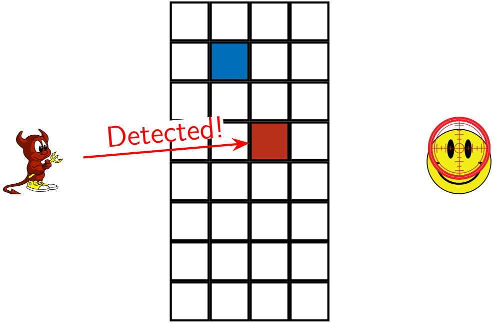
Check if corresponding cache line from step 2 has been loaded by the victim's program

Cache (8 sets, 4 ways)



Flush + Reload

Cache (8 sets, 4 ways)



Detected!

Cache attacks

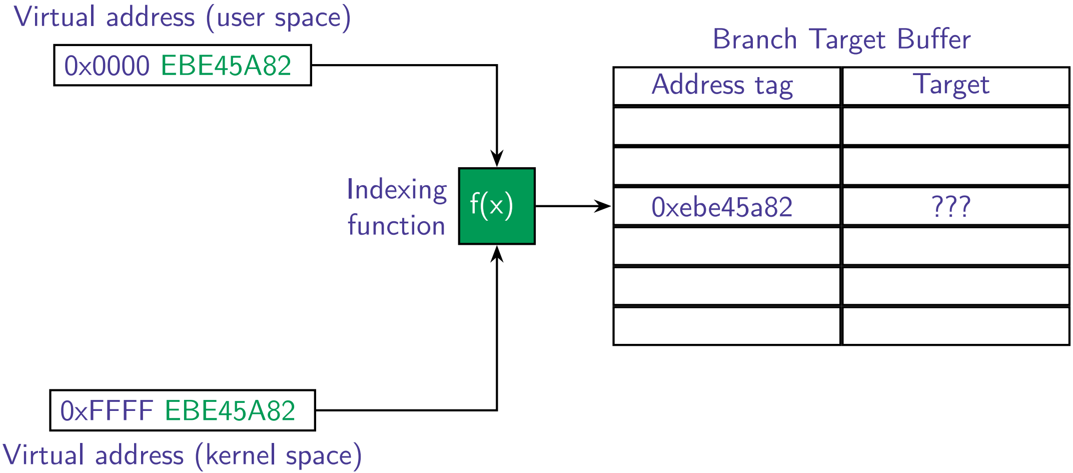
- Evict + Time
- Prime + Probe
- Prime + Abort
- Flush + Flush
- Evict + Reload
- AnC (ASLR $\oplus$ Cache)
- and etc.

Agenda

Basic attacks

Branch-prediction attacks

Branch-prediction attacks



Virtual address (user space)

0x0000 EBE45A82

Indexing function

f(x)

0xFFFF EBE45A82

Virtual address (kernel space)

Branch Target Buffer

Address tag	Target
0xebe45a82	???

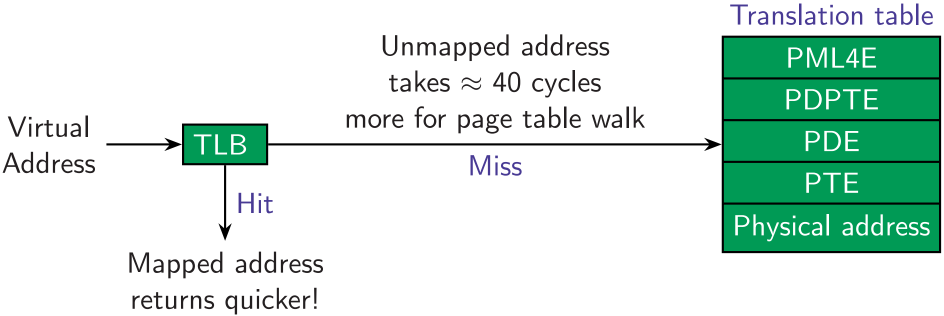
Branch Target Buffer addressing scheme in Haswell processor

Agenda

Basic attacks

TLB-based attacks

TLB-based attacks


Translation table
Unmapped address takes ≈ 40 cycles more for page table walk
Virtual Address
TLB
Miss
PML4E
PDPTE
PDE
PTE
Physical address
Hit
Mapped address returns quicker!

A translation lookaside buffer (TLB) is a memory cache that is used to reduce the time taken to access a user memory location

Agenda

Basic attacks

Exception-based attacks

Exception-based attacks

- Scheduler interrupts
- Instruction aborts
- Page faults
- Behavioral differences (e.g, error code)

Agenda

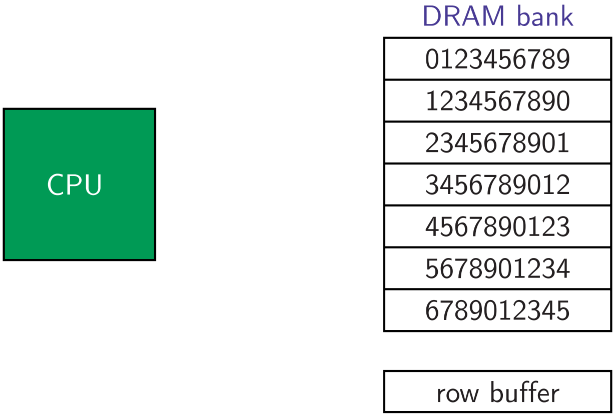
Basic attacks

DRAM-based attacks

Reading from DRAM

Complex DRAM-based attacks

Reading from DRAM



CPU

DRAM bank

0123456789

1234567890

2345678901

3456789012

4567890123

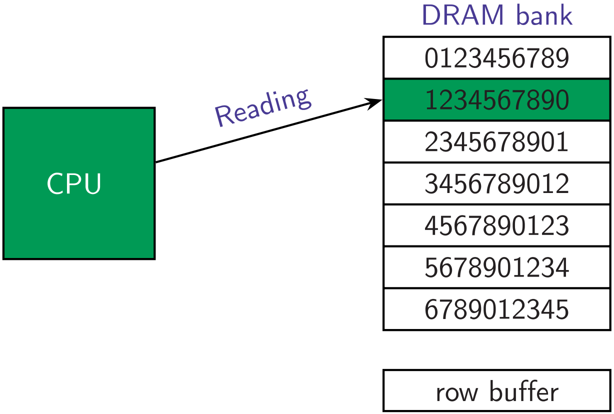
5678901234

6789012345

row buffer

Reading from DRAM

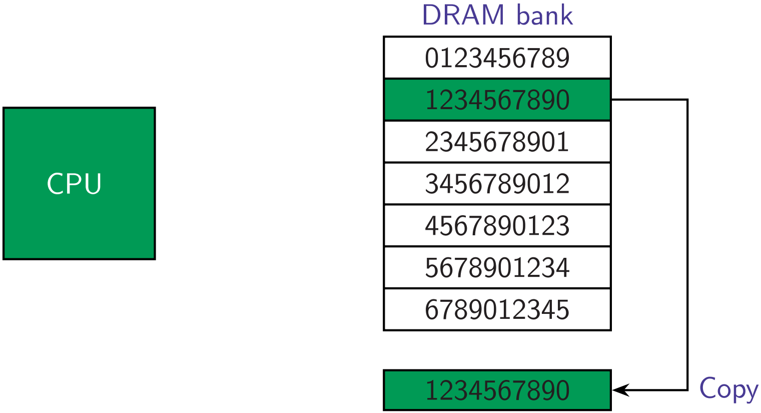
Reading from DRAM



0123456789
1234567890
2345678901
3456789012
4567890123
5678901234
6789012345

CPU reads row 1, row buffer empty

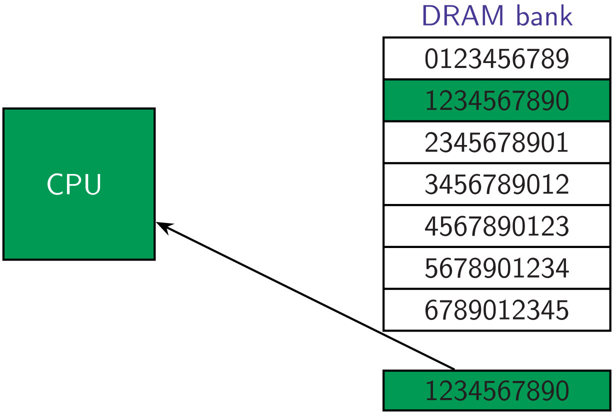
Reading from DRAM



0123456789
1234567890
2345678901
3456789012
4567890123
5678901234
6789012345

1234567890

Reading from DRAM



CPU

DRAM bank

0123456789

1234567890

2345678901

3456789012

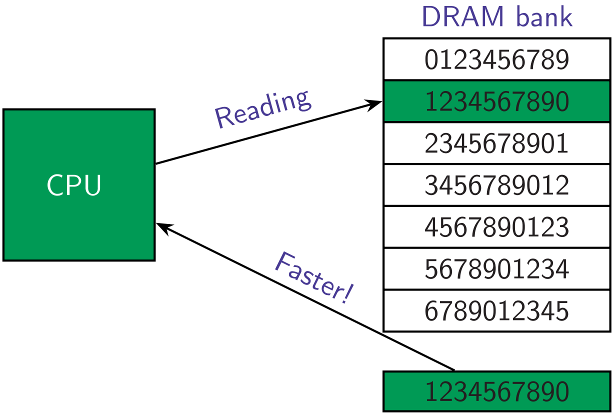
4567890123

5678901234

6789012345

1234567890

Reading from DRAM



DRAM bank

0123456789

1234567890

2345678901

3456789012

4567890123

5678901234

6789012345

CPU

Reading

Faster!

1234567890

CPU reads row 1, row buffer now full

Complex DRAM-based attacks

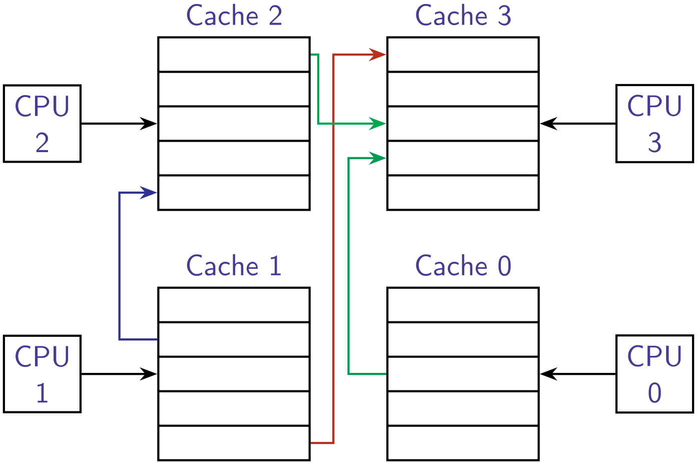
- DRAMA
- Row hit (Flush + Reload)
- Row miss (Prime + Probe)
- and etc.

Agenda

Basic attacks

Covert channels

Covert channels



Cross-core covert channels

Covert channels

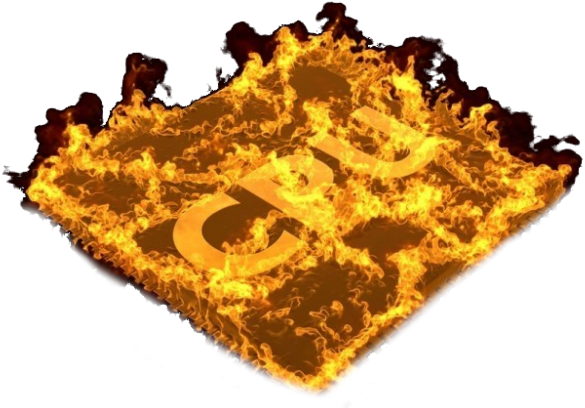
- Cache-based covert channels (shared libraries)
- Row miss attack (DRAM)
- Thermal covert channels
- Radio covert channels

Agenda

Software-based Microarchitectural Fault Attacks

Rowhammer

Software-based Microarchitectural Fault Attacks



Software-based microarchitectural fault attacks do not require physical access, but instead only some form of code execution on the target system

Agenda

Software-based Microarchitectural Fault Attacks

Rowhammer

Rowhammer. Exploitation primitives

Variations of Rowhammer

Rowhammer. Exploitation primitives

- Fast uncached memory access
- Physical memory massaging
- Physical memory addressing

Variations of Rowhammer

- Flip Feng Shui — targeted Rowhammer
- Throwhammer — remote Rowhammer
- Nethammer — better remote Rowhammer
- Drammer, RAMpage — exploitation ARM-based hardware
- Glitch — better exploitation ARM-based hardware

Agenda

Meltdown & Spectre

- Derived attacks and not only
- Abstract example of exploitation

Agenda

Meltdown & Spectre

Derived attacks and not only

Derived attacks and not only

Spectre-NG

- MeltdownPrime & SpectrePrime
- SgxPectre
- SMM Speculative Execution Attacks
- BranchScope
- LazyFP
- ...

Derived attacks and not only

- Spectre 1.1, 1.2 (Speculative Buffer Overflows)
- SpectreRSB
- NetSpectre
- L1TF (Foreshadow)
- and etc.

TotalMeltdown? and other patches...

Agenda

Meltdown & Spectre

Abstract example of exploitation

Abstract example of exploitation

The four components of speculation techniques

1. Speculation primitive

Abstract example of exploitation

The four components of speculation techniques

1. Speculation primitive

- Bypass out of bounds checks
- Training of branch predictor
- Speculatively read an earlier value of the data
- Pending exceptions
- Exploit branch history table
- Exploit the Return Stack Buffer
- Speculatively write to register (buffer overflow)

} Microarchitecture — ?

Abstract example of exploitation

Type of BP

Algorithm of BP

Environment of BP

Foundation of tower
speculative-based attack

Abstract example of exploitation

The four components of speculation techniques

1. Speculation primitive
2. Windowing gadget

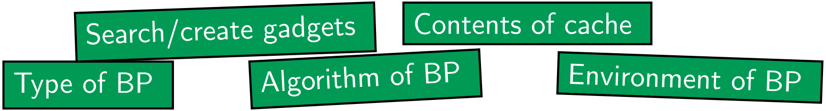
Abstract example of exploitation

The four components of speculation techniques

1. Speculation primitive
2. Windowing gadget

- Non-cached loads
- Dependency chain of loads
- Dependency chain of integer ALU operations

Abstract example of exploitation



Search/create gadgets

Contents of cache

Type of BP

Algorithm of BP

Environment of BP

Tower
speculative-based attack

Abstract example of exploitation

The four components of speculation techniques

1. Speculation primitive
2. Windowing gadget
3. Disclosure gadget

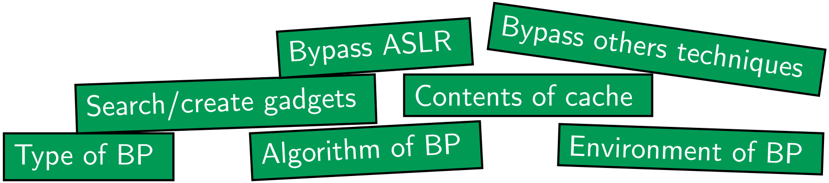
Abstract example of exploitation

The four components of speculation techniques

1. Speculation primitive
2. Windowing gadget
3. Disclosure gadget

- ASLR
- CFI
- SMAP
- DEP/NX
- retpoline
- and others.

Abstract example of exploitation



Bypass ASLR

Bypass others techniques

Search/create gadgets

Contents of cache

Type of BP

Algorithm of BP

Environment of BP

Babel tower
speculative-based attack

Abstract example of exploitation

The four components of speculation techniques

1. Speculation primitive
2. Windowing gadget
3. Disclosure gadget
4. Disclosure primitive

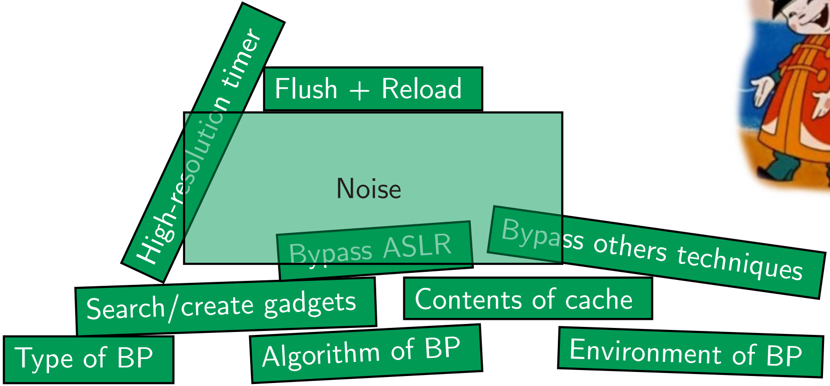
Abstract example of exploitation

The four components of speculation techniques

1. Speculation primitive
2. Windowing gadget
3. Disclosure gadget
4. Disclosure primitive

- Architecture of cache
- Replacement policies
- Exclusive and inclusive
- Type of cache attack
- Noise
- High-resolution timer
- and etc.

Abstract example of exploitation



High-resolution timer

Flush + Reload

Noise

Bypass ASLR

Bypass others techniques

Search/create gadgets

Contents of cache

Type of BP

Algorithm of BP

Environment of BP

Babel tower

speculative-based attack

Agenda

Summary

Summary

- Software-based microarchitectural attacks become **a very popular**

Summary

- Software-based microarchitectural attacks become **a very popular**
- Requires **a lot of resources** to develop working exploit

Summary

- Software-based microarchitectural attacks become **a very popular**
- Requires **a lot of resources** to develop working exploit
- Microarchitectural attacks may be **automated**

Summary

- Software-based microarchitectural attacks become **a very popular**
- Requires **a lot of resources** to develop working exploit
- Microarchitectural attacks may be **automated**
- Many attacks have **not yet been published**

Summary

- Software-based microarchitectural attacks become **a very popular**
- Requires **a lot of resources** to develop working exploit
- Microarchitectural attacks may be **automated**
- Many attacks have **not yet been published**
- Countermeasures come with a **performance impact**

Questions?

References I

- Daniel Gruss
  Software-based Microarchitectural Attacks.

- Moritz Lipp, Daniel Gruss
  ARMageddon: Cache Attacks on Mobile Devices.

- D. Page
  MASCAB: a Micro-Architectural Side-Channel Attack Bibliography.

- Pessl P., Gruss D. and others
  DRAMA: Exploiting DRAM Addressing for Cross-CPU Attacks.

- Bos H., Fratantonio Y. and others
  Drammer: Determenistic Rowhammer Attacks on Mobile Platforms.

- Microsoft
  Mitigating speculative execution side channel hardware vulnerabilities.

References II

- Google Project Zero
  Reading privileged memory with a side-channel.

- Daniel Gruss, Moritz Lipp
  KASLR is Dead: Long Live KASLR.

- Daniel Gruss, Clémentine Maurice and others
  Flush+Flush: A Fast and Stealthy Cache Attack.

- Fangfei Liu, Yuval Yarom and others
  Last-Level Cache Side-Channel Attacks are Practical.

- Caroline Trippel, Daniel Lustig, Margaret Martonosi
  MeltdownPrime and SpectrePrime: Automatically-Synthesized Attacks Exploiting Invalidation-Based Coherence Protocols.

References III

- Michael Schwarz, Clémentine Maurice, Daniel Gruss, Stefan Mangard
  Fantastic Timers and Where to Find Them: High-Resolution Microarchitectural Attacks in JavaScript.

- Moritz Lipp, Misiker Tadesse Aga and others
  Nethammer: Inducing Rowhammer Faults through Network Requests.

- Andrei Tatar, Radhesh Krishnan and others
  Throwhammer: Rowhammer Attacks over the Network and Defenses.

- Giovanni Camurati, Sebastian Poeplau and others
  Screaming Channels: When Electromagnetic Side Channels Meet Radio Transceivers.

- Julian Stecklina, Thomas Prescher
  LazyFP: Leaking FPU Register State using Microarchitectural Side-Channels.

- Mordechai Guri, Assaf Kachlon and others
  GSMem: Data Exfiltration from Air-Gapped Computers over GSM Frequencies.

References IV

- Dean Sullivan, Orlando Arias, Travis Meade, Yier Jin
  Microarchitectural Minefields: 4K-Aliasing Covert Channel and Multi-Tenant Detection in IaaS Clouds.
- Gras B., Razavi K., Bosman E., Bos H., Giuffrida C.
  ASLR on the Line: Practical Cache Attacks on the MMU.
- van Schaik S., Giuffrida C., Bos H., Razavi K.
  Malicious Management Unit: Why Stopping Cache Attacks in Software is Harder Than You Think.
- Daniel Gruss, Anders Fogh and others
  Prefetch Side-Channel Attacks: Bypassing SMAP and Kernel ASLR.
- Esmaeil Mohammadian Koruyeh, Khaled N. Khasawneh and others
  Spectre Returns! Speculation Attacks using the Return Stack Buffer.

References V

- Giorgi Maisuradze, Christian Rossow
  ret2spec: Speculative Execution Using Return Stack Buffers.

- Guoxing Chen, Sanchuan Chen and others
  SgxPectre Attacks: Leaking Enclave Secrets via Speculative Execution.

- Moritz Lipp, Michael Schwarz and others
  Meltdown.

- Paul Kocher, Daniel Genkin and others
  Spectre Attacks: Exploiting Speculative Execution.

- ARM Whitepaper
  Cache Speculation Side-channels.

- Michael Schwarz, Martin Schwarzl, Moritz Lipp, Daniel Gruss
  NetSpectre: Read Arbitrary Memory over Network.

References VI

Sophia D'Antoine
Out-of-Order Execution and Its Applications.

Vladimir Kiriansky, Carl Waldspurger
Speculative Buffer Overflows: Attacks and Defenses.

Gras B., Razavi K., Bos H., Giuffrida C.
Translation Leak-aside Buffer: Defeating Cache Side-channel Protections with TLB Attacks.

Craig Disselkoen, David Kohlbrenner, Leo Porter, Dean Tullsen
Prime+Abort: A Timer-Free High-Precision L3 Cache Attack using Intel TSX.

Moritz Lipp, Michael Schwarz
Meltdown & Spectre Side-channels considered hARMful.

Jon Masters
Exploiting modern microarchitectures: Meltdown, Spectre, and other attacks.

References VII

- Moritz Lipp
  Cache attacks on ARM.
- Evtyushkin D., Ponomarev D., Abu-Ghazaleh N.
  Jump over ASLR: attacking branch predictors to bypass ASLR.