Software-based Microarchitectural Attacks: What do we learn from Meltdown and Spectre?

Daniel Gruss

March 26, 2019

Graz University of Technology

You realize it is something big when...

You realize it is something big when...

- it is in the news, all over the world


FOX BUSINESS
WASHINGTON, D.C.
WASHINGTON, D.C.
WINTER STORM
NEWS ALERT
INTEL REVEALS DESIGN FLAW THAT
COULD ALLOW HACKERS TO ACCESS DATA
FOX BUSINESS NETWORK
@FOXBUSINESS


DEVELOPING STORY
COMPUTER CHIP FLAWS IMPACT BILLIONS OF DEVICES
LIVE
CNN
DAX ▲164.69
NEWS STREAM


GLOBAL
COMPUTER CHIP SCARE
The bugs are known as 'Spectre' and 'Meltdown'
BBC WORLD NEWS
£:HKS 10.58
EURO:£ 0.891


ESEARC
SECURITY FLAW REVEALED
Intel (Prev)
45.26 -1.59 [-3.39%]
Intel (After Hours)
44.85 -0.41 [-0.91%]
CAPITAL CONNECTION
SHROUT: ISSUE NOT UNIQUE TO INTEL, BUT IT'S AFFECTED THE MOST
CNBC

ABO SHOP AKADEMIE JOBS MEHR ▾ E-PAPER AUDIO APPS ARCHIV ANMELDEN

ZEIT ONLINE

Suche

Politik Gesellschaft Wirtschaft Kultur ▾ Wissen **Digital** Campus ▾ Arbeit Entdecken Sport ZEITmagazin mehr ▾ #D18

AUS DER SERIE
Was bewegt

Daniel Gruss

Der Kernschmelzer

Daniel Gruss hat eine schwere Sicherheitslücke in Computerchips entdeckt. Warum gelingt dem Informatiker, woran die Hersteller scheitern?

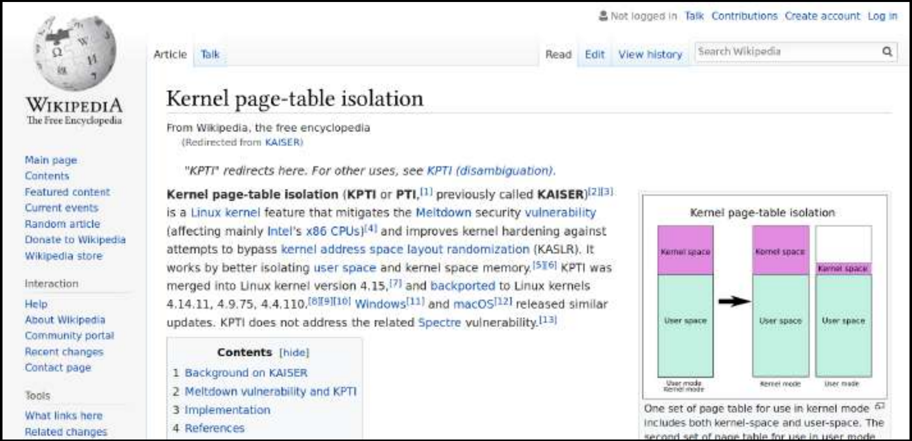
Von **Jens Tönnesmann**

7. März 2018, 16:46 Uhr / Editiert am 9. März 2018, 20:11 Uhr / 26 Kommentare

AUS DER
ZEIT NR. 11/2018

You realize it is something big when...

- it is in the news, all over the world
- you get a Wikipedia article in multiple languages



Not logged in Talk Contributions Create account Log in

Article Talk Read Edit View history Search Wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

What links here
Related changes

Kernel page-table isolation

From Wikipedia, the free encyclopedia

(Redirected from KAISER)

"KPTI" redirects here. For other uses, see KPTI (disambiguation).

Kernel page-table isolation (**KPTI** or **PTI**,[1] previously called **KAISER**)[2][3] is a Linux kernel feature that mitigates the Meltdown security vulnerability (affecting mainly Intel's x86 CPUs)[4] and improves kernel hardening against attempts to bypass kernel address space layout randomization (KASLR). It works by better isolating user space and kernel space memory.[5][6] KPTI was merged into Linux kernel version 4.15,[7] and backported to Linux kernels 4.14.11, 4.9.75, 4.4.110.[8][9][10] Windows[11] and macOS[12] released similar updates. KPTI does not address the related Spectre vulnerability.[13]

Contents [hide]

One set of page table for use in kernel mode includes both kernel-space and user-space. The second set of page table for use in user mode

Not logged in Talk Contributions Create account Log in

WIKIPEDIA
The Free Encyclopedia

Article Talk Read Edit View history Search Wikipedia

Meltdown (security vulnerability)

From Wikipedia, the free encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

What links here
Related changes
Upload file

Meltdown is a hardware vulnerability affecting Intel x86 microprocessors and some ARM-based microprocessors.[1][2][3] It allows a rogue process to read all memory, even when it is not authorized to do so.

Meltdown affects a wide range of systems. At the time of disclosure, this included all devices running any but the most recent and patched versions of iOS,[4] Linux[5][6], macOS,[4] or Windows. Accordingly, many servers and cloud services were impacted,[7] as well as a potential majority of smart devices and embedded devices using ARM based processors (mobile devices, smart TVs and others), including a wide range of networking equipment. A purely software workaround to Meltdown has been assessed as slowing computers between 5 and 30 percent in certain specialized workloads,[8] although companies responsible for software correction of the exploit are reporting minimal impact from general benchmark testing.[9]

Meltdown was issued a Common Vulnerabilities and Exposures ID of CVE-2017-5754, also known as *Rogue Data Cache Load*,[2] in January 2018. It was disclosed in conjunction with another exploit, Spectre, with which it shares some, but not all characteristics. The Meltdown and Spectre vulnerabilities are considered "catastrophic"

MELTDOWN

The logo used by the team that discovered the vulnerability

Not logged in Talk Contributions Create account Log in

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

What links here
Related changes
Upload file

Article Talk Read Edit View history Search Wikipedia

Spectre (security vulnerability)

From Wikipedia, the free encyclopedia

Spectre is a vulnerability that affects modern microprocessors that perform branch prediction.[1][2][3] On most processors, the speculative execution resulting from a branch misprediction may leave observable side effects that may reveal private data to attackers. For example, if the pattern of memory accesses performed by such speculative execution depends on private data, the resulting state of the data cache constitutes a side channel through which an attacker may be able to extract information about the private data using a timing attack.[4][5][6]

Two Common Vulnerabilities and Exposures IDs related to Spectre, CVE-2017-5753 (bounds check bypass) and CVE-2017-5715 (branch target injection), have been issued.[7] JIT engines used for JavaScript were found vulnerable. A website can read data stored in the browser for another website, or the browser's memory itself.[8]

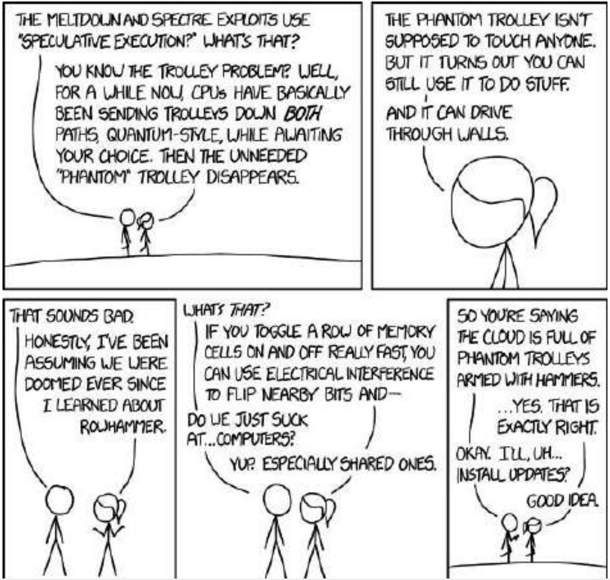
SPECTRE

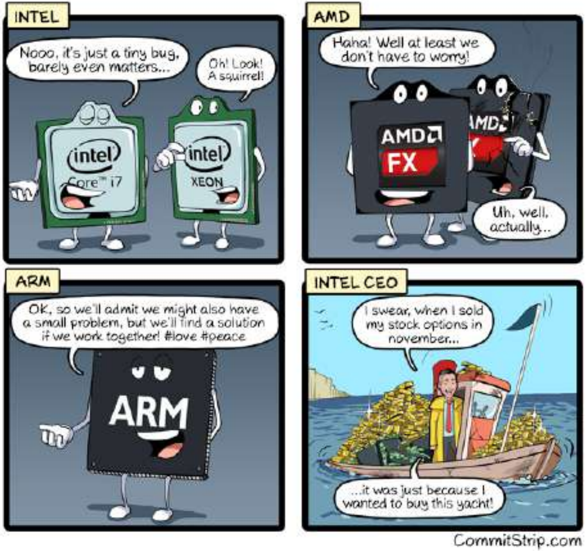
A logo created for the vulnerability, featuring a ghost with a branch

Several procedures to help protect home computers and related devices from the Spectre (and Meltdown) security vulnerabilities have been published.[9][10][11][12] Spectre patches have been reported to significantly slow down performance, especially on older computers; on the newer 8th generation Core platforms, benchmark performance drops of 2–14 percent have been measured.[13] Meltdown patches may also produce performance loss.[5][14][15] On January 18, 2018, unwanted reboots, even for newer Intel chips, due to

You realize it is something big when...

- it is in the news, all over the world
- you get a Wikipedia article in multiple languages
- there are comics, including xkcd

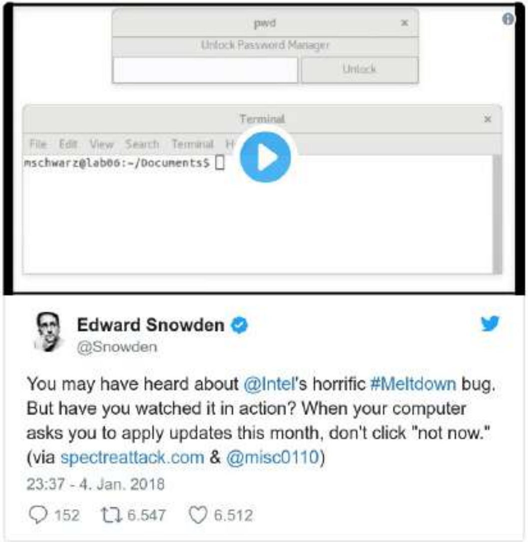

THE MELTDOWN AND SPECTRE EXPLOITS USE "SPECULATIVE EXECUTION?" WHAT'S THAT?
YOU KNOW THE TROLLEY PROBLEM? WELL, FOR A WHILE NOW, CPUs HAVE BASICALLY BEEN SENDING TROLLEYS DOWN *BOTH* PATHS, QUANTUM-STYLE, WHILE AWAITING YOUR CHOICE. THEN THE UNNEEDED "PHANTOM" TROLLEY DISAPPEARS.
THE PHANTOM TROLLEY ISN'T SUPPOSED TO TOUCH ANYONE. BUT IT TURNS OUT YOU CAN STILL USE IT TO DO STUFF.
AND IT CAN DRIVE THROUGH WALLS.
THAT SOUNDS BAD.
HONESTLY, I'VE BEEN ASSUMING WE WERE DOOMED EVER SINCE I LEARNED ABOUT ROWHAMMER.
WHAT'S *THAT?*
IF YOU TOGGLE A ROW OF MEMORY CELLS ON AND OFF REALLY FAST, YOU CAN USE ELECTRICAL INTERFERENCE TO FLIP NEARBY BITS AND—
DO WE JUST SUCK AT...COMPUTERS?
YUP. ESPECIALLY SHARED ONES.
SO YOU'RE SAYING THE CLOUD IS FULL OF PHANTOM TROLLEYS ARMED WITH HAMMERS.
...YES. THAT IS EXACTLY RIGHT.
OKAY. I'LL, UH... INSTALL UPDATES?
GOOD IDEA.


INTEL
Nooo, it's just a tiny bug, barely even matters...
Oh! Look! A squirrel!
AMD
Haha! Well at least we don't have to worry!
Uh, well, actually...
ARM
Ok, so we'll admit we might also have a small problem, but we'll find a solution if we work together! #love #peace
INTEL CEO
I swear, when I sold my stock options in november...
...it was just because I wanted to buy this yacht!
CommitStrip.com

The Fallout

You realize it is something big when...

- it is in the news, all over the world
- you get a Wikipedia article in multiple languages
- there are comics, including xkcd
- you get a lot of Twitter follower after Snowden mentioned you


pwd
Unlock Password Manager
Unlock
Terminal
File Edit View Search Terminal
mschwarz@lab06:~/Documents$
Edward Snowden
@Snowden
You may have heard about @Intel's horrific #Meltdown bug. But have you watched it in action? When your computer asks you to apply updates this month, don't click "not now." (via spectreattack.com & @misc0110)
23:37 - 4. Jan. 2018
152
6.547
6.512

1337 4242

FOOD CACHE

Revolutionary concept!

Store your food at home, never go to the grocery store during cooking.

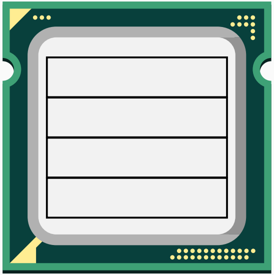
Can store **ALL** kinds of food.

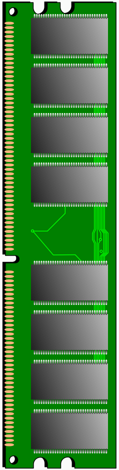
ONLY TODAY INSTEAD OF ~~$1,300~~

$1,299

ORDER VIA PHONE: +555 12345

```
printf("%d", i);
printf("%d", i);
```

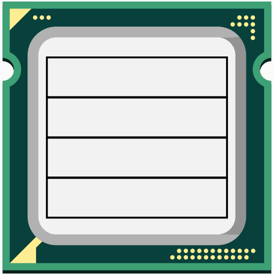


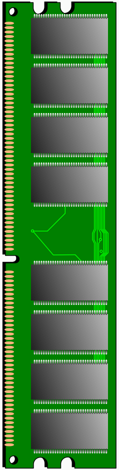


CPU Cache

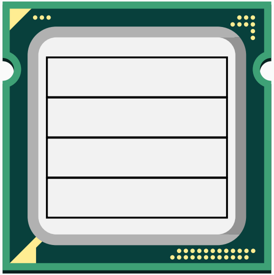
```
printf("%d", i);
printf("%d", i);
```

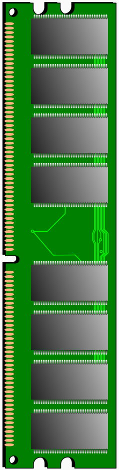
Cache miss



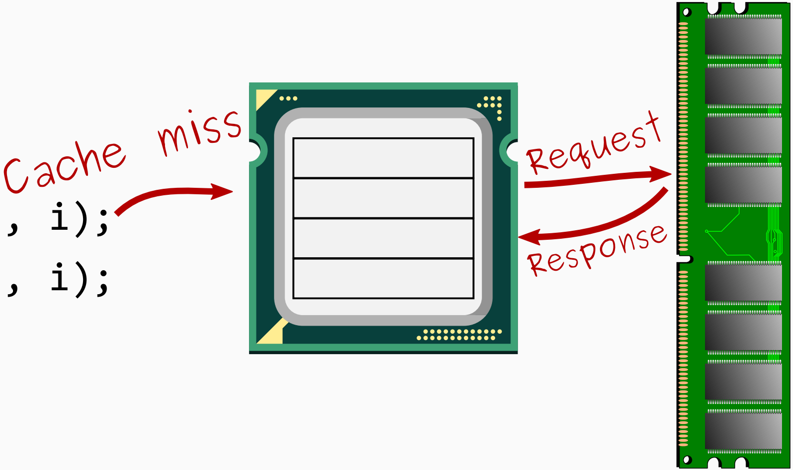


```
printf("%d", i);
printf("%d", i);
```

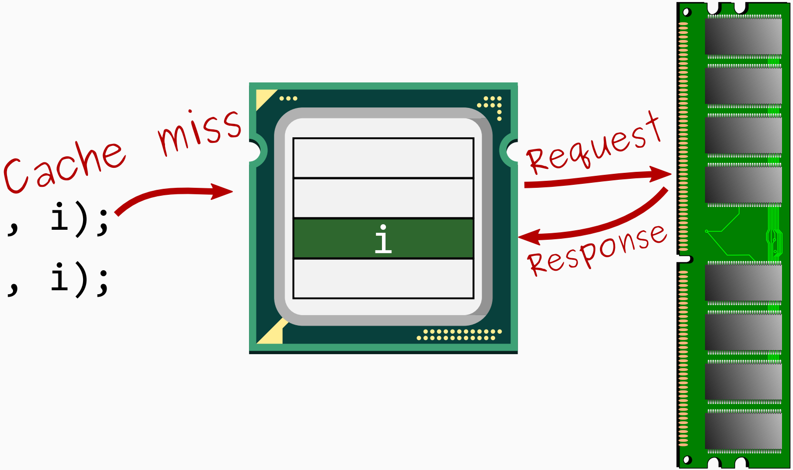




```c
printf("%d", i);
printf("%d", i);
```

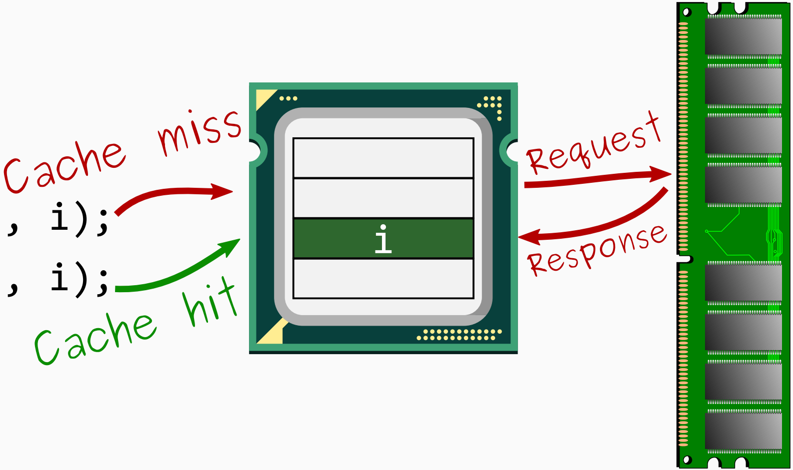


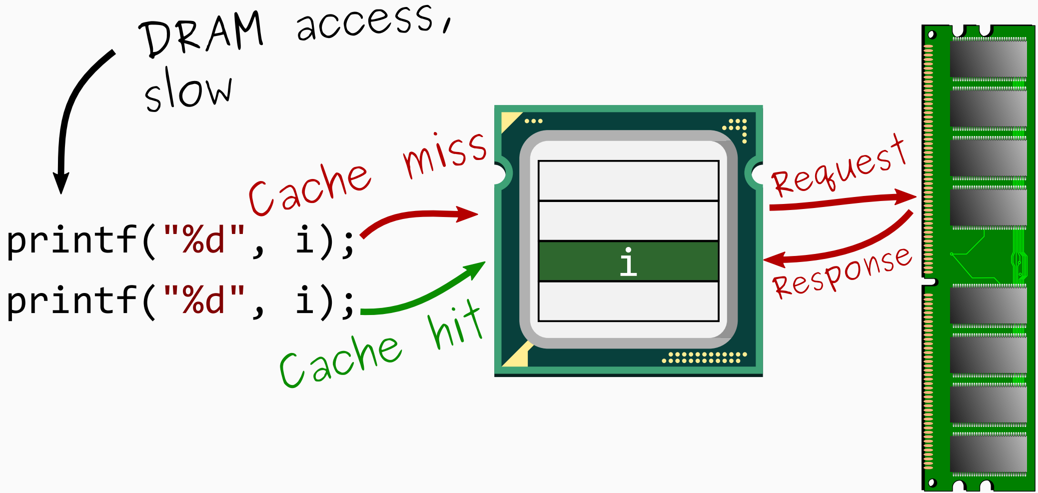
```
printf("%d", i);
printf("%d", i);
```



CPU Cache

```
printf("%d", i);
printf("%d", i);
```



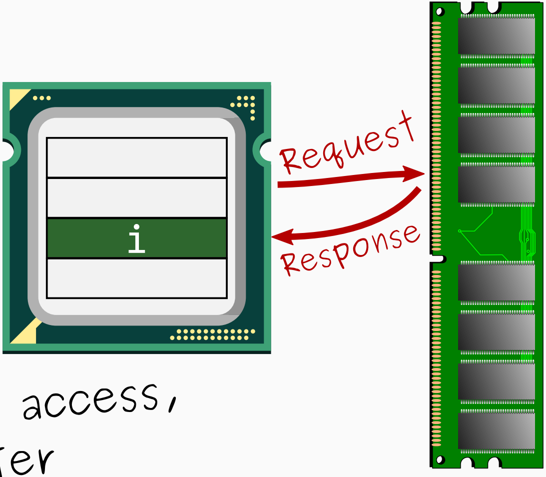

DRAM access,
slow
printf("%d", i);
printf("%d", i);
Cache miss
Cache hit
i
Request
Response

DRAM access, slow

```
printf("%d", i);
printf("%d", i);
```

Cache miss

Cache hit

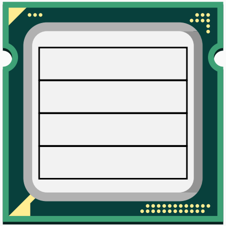


No DRAM access, much faster

Shared Memory

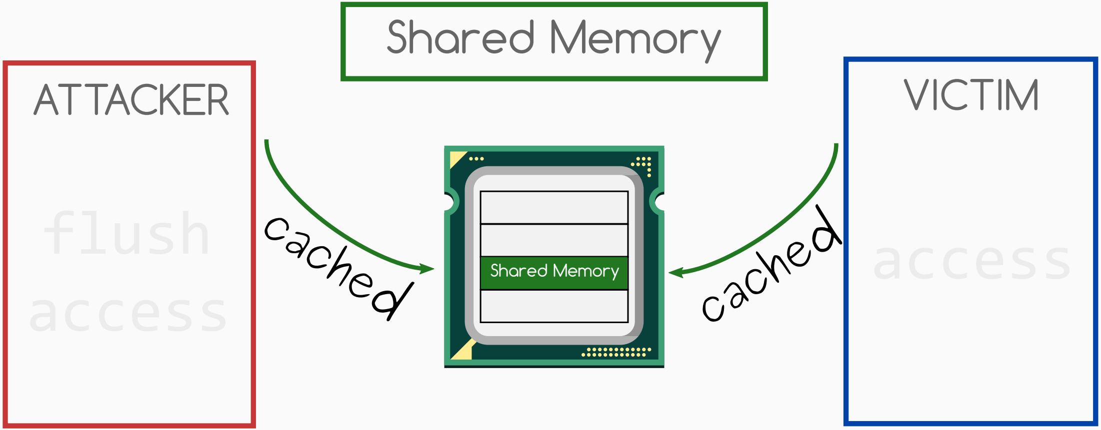
ATTACKER

flush
access

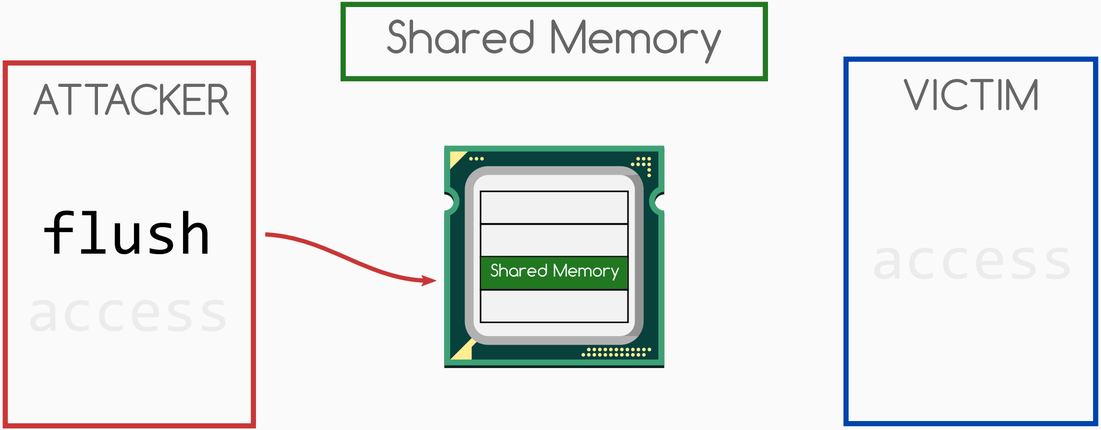


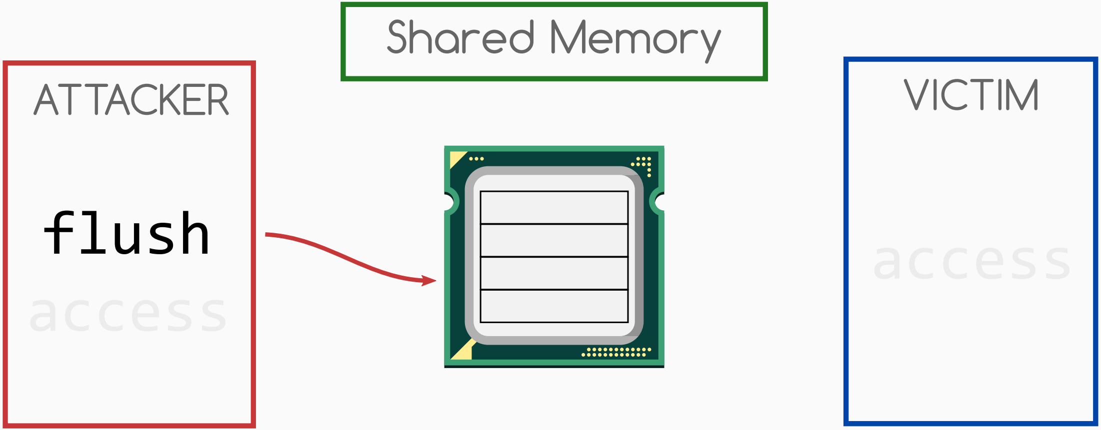
VICTIM

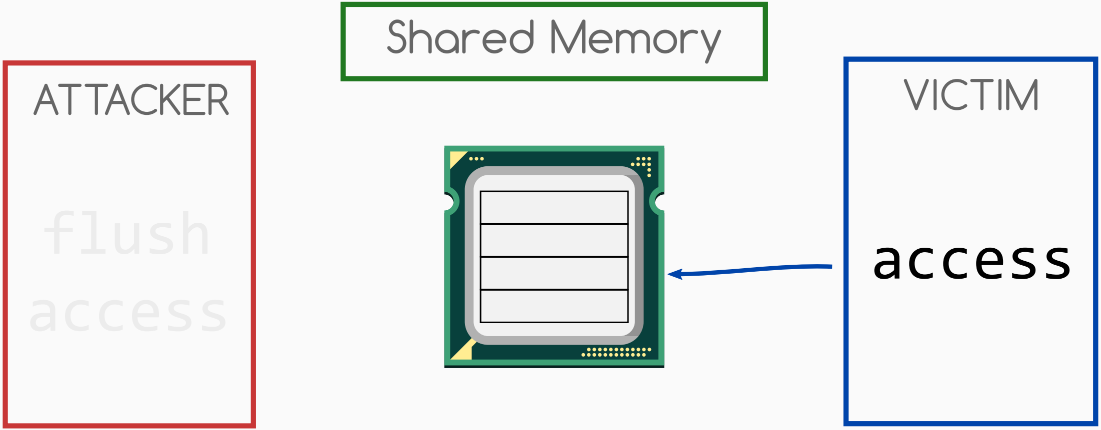
access



Shared Memory
ATTACKER
flush
access
cached
Shared Memory
cached
VICTIM
access


Shared Memory
ATTACKER
flush
access
Shared Memory
VICTIM
access


Shared Memory
ATTACKER
flush
access
VICTIM
access

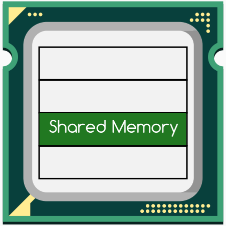


Shared Memory
ATTACKER
flush
access
VICTIM
access

Shared Memory

ATTACKER

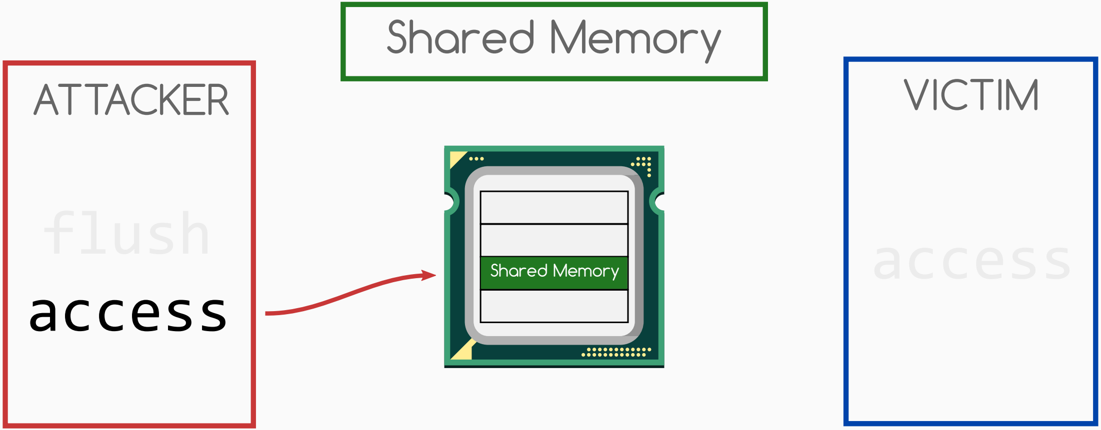
flush
access



VICTIM

access

Flush+Reload

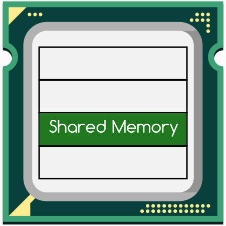


Shared Memory

ATTACKER

flush

access

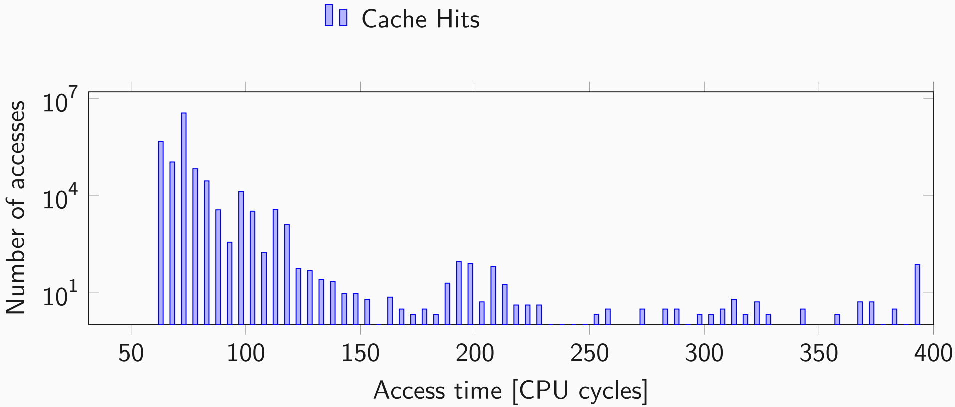


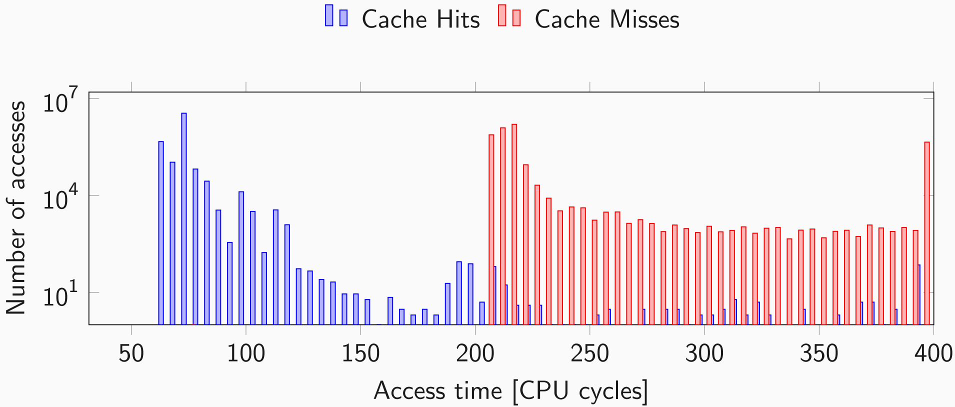
VICTIM

access

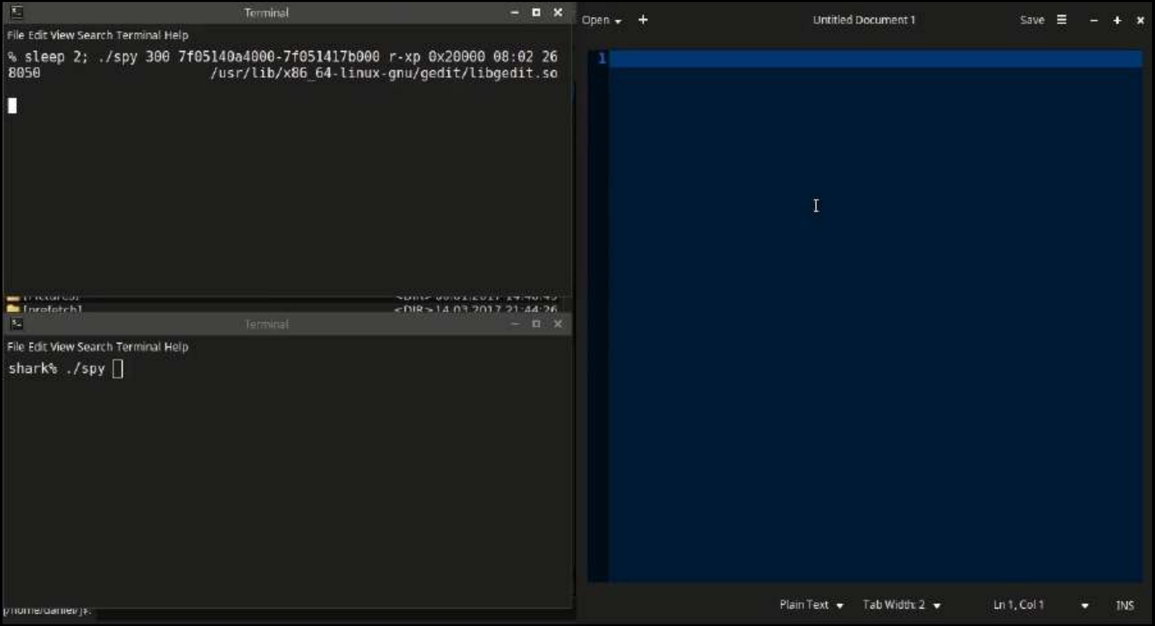
fast if victim accessed data,
slow otherwise

Memory Access Latency





Cache Hits
Cache Misses
Number of accesses
10^7
10^4
10^1
50
100
150
200
250
300
350
400
Access time [CPU cycles]



Cache Template Attack Demo


local — Konsole
File Edit View Bookmarks Settings Help
local
sender (ec2)
receiver (ec2)
Active Interface: eth0
Interface Speed: unknown
HELLO FROM THE OTHER SIDE (DEMO):
VIDEO STREAMING OVER CACHE COVERT CHANNEL
Line 17, Column 1
INSERT
UTF-8
Normal


Back to Work

7. Serve with cooked and peeled potatoes

Wait for an hour

Wait for an hour

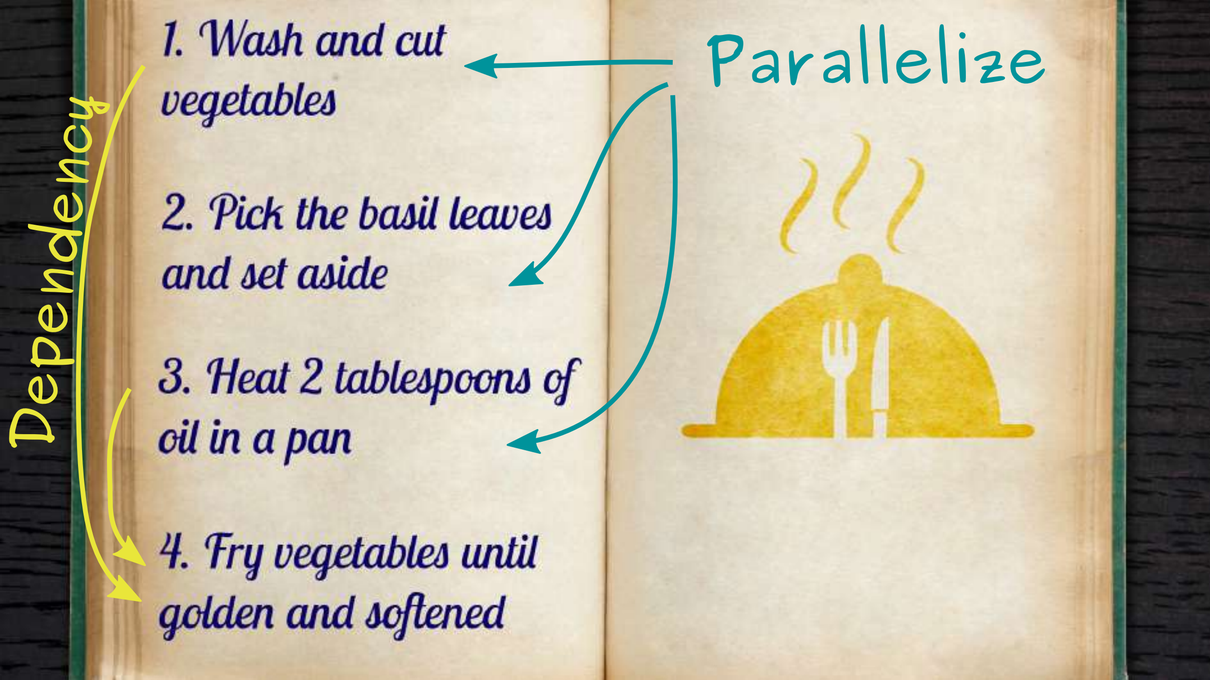
LATENCY

1. Wash and cut vegetables

2. Pick the basil leaves and set aside

3. Heat 2 tablespoons of oil in a pan

4. Fry vegetables until golden and softened


1. Wash and cut vegetables
2. Pick the basil leaves and set aside
3. Heat 2 tablespoons of oil in a pan
4. Fry vegetables until golden and softened
Parallelize
Dependency

```c
int width = 10, height = 5;

float diagonal = sqrt(width * width
                      + height * height);
int area = width * height;

printf("Area %d x %d = %d\n", width, height, area);
```


Parallelize

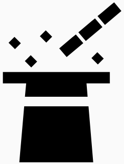
Dependency

```
int width = 10, height = 5;

float diagonal = sqrt(width * width
                    + height * height);
int area = width * height;

printf("Area %d x %d = %d\n", width, height, area);
```


```c
char data = *(char*)0xffffffff81a000e0;
printf("%c\n", data);
```

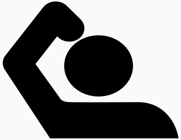
```
char data = *(char*)0xffffffff81a000e0;
printf("%c\n", data);
```

```
segfault at ffffffff81a000e0 ip
   0000000000400535
      sp 00007ffce4a80610 error 5 in reader
```

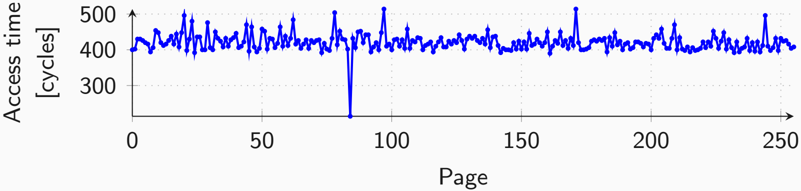


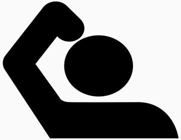
Adapted code

```
*(volatile char*)0;
array[84 * 4096] = 0; // unreachable
```

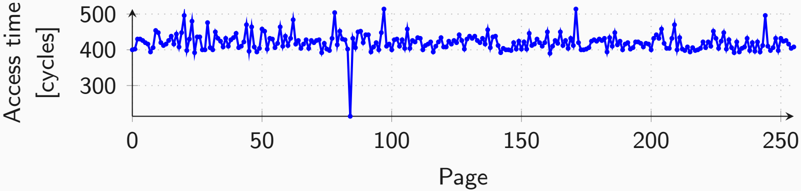


Flush+Reload over all pages of the array





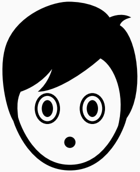
Flush+Reload over all pages of the array



This also works on AMD and ARM!

- Combine the two things

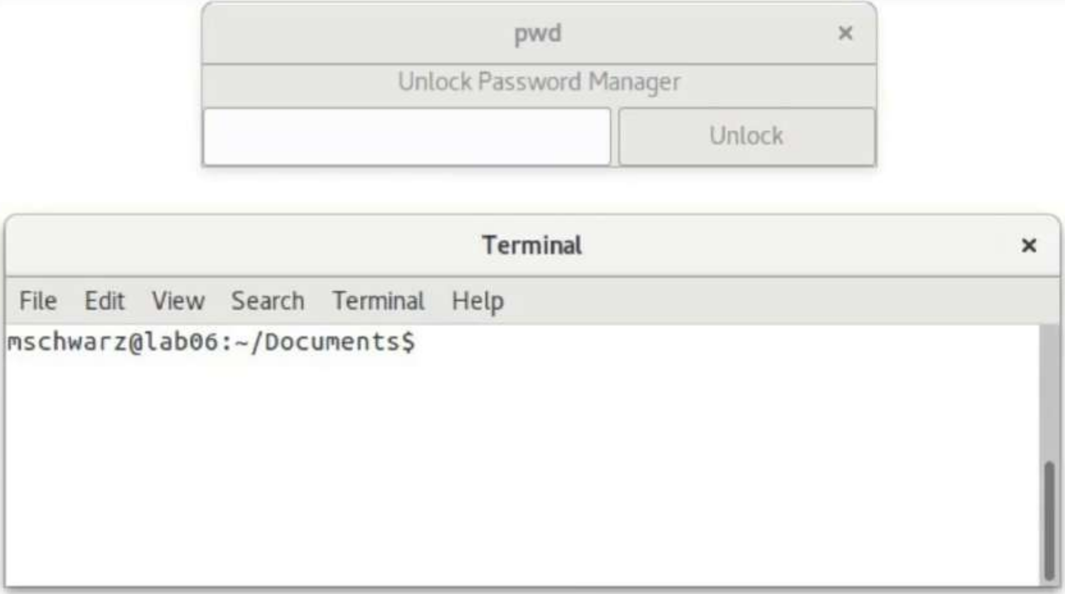
```
char data = *(char*)0xffffffff81a000e0;
array[data * 4096] = 0;
```



Flush+Reload again...

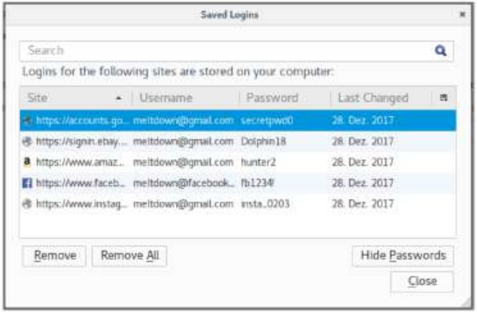
... Meltdown actually works.


I SHIT YOU NOT
THERE WAS KERNEL MEMORY ALL OVER THE TERMINAL


pwd
Unlock Password Manager
Unlock
Terminal
File Edit View Search Terminal Help
mschwarz@lab06:~/Documents$


CAN YOU
ENHANCE THAT

```
meltdown@meltdown ~/ppm2 % taskset 1 ./imgdump 0x375a00000 14919 > outp
ut.flif
Reading from 0xffff880375a00000
```




```
f94b7690: e5 e5 e5 e5 e5 e5 e5 e5 e5  e5 e5 e5 e5 e5 e5 e5 |................|
f94b76a0: e5 e5 e5 e5 e5 e5 e5 e5 e5  e5 e5 e5 e5 e5 e5 e5 |................|
f94b76b0: 70 52 b8 6b 96 7f XX XX XX  XX XX XX XX XX XX XX |pR.k............|
f94b76c0: 09 XX XX XX XX XX XX XX XX  XX XX XX XX XX XX XX |................|
f94b76d0: XX XX XX XX XX XX XX XX XX  XX XX XX XX XX XX XX |................|
f94b76e0: XX XX XX XX XX XX XX XX XX  XX XX XX XX XX XX 81 |................|
f94b76f0: 12 XX e0 81 19 XX e0 81 44  6f 6c 70 68 69 6e 31 |........Dolphin1|
f94b7700: 38 e5 e5 e5 e5 e5 e5 e5 e5  e5 e5 e5 e5 e5 e5 e5 |8...............|
f94b7710: 70 52 b8 6b 96 7f XX XX XX  XX XX XX XX XX XX XX |pR.k............|
f94b7720: XX XX XX XX XX XX XX XX XX  XX XX XX XX XX XX XX |................|
f94b7730: XX XX XX XX 4a XX XX XX XX  XX XX XX XX XX XX XX |....J...........|
f94b7740: XX XX XX XX XX XX XX XX XX  XX XX XX XX XX XX XX |................|
f94b7750: XX XX XX XX XX XX XX XX XX  XX e0 81 69 6e 73 74 |............inst|
f94b7760: 61 5f 30 32 30 33 e5 e5 e5  e5 e5 e5 e5 e5 e5 e5 |a_0203..........|
f94b7770: 70 52 18 7d 28 7f XX XX XX  XX XX XX XX XX XX XX |pR.}(...........|
f94b7780: XX XX XX XX XX XX XX XX XX  XX XX XX XX XX XX XX |................|
f94b7790: XX XX XX XX 54 XX XX XX XX  XX XX XX XX XX XX XX |....T...........|
f94b77a0: XX XX XX XX XX XX XX XX XX  XX XX XX XX XX XX XX |................|
f94b77b0: XX XX XX XX XX XX XX XX XX  XX XX XX 73 65 63 72 |............secr|
f94b77c0: 65 74 70 77 64 30 e5 e5 e5  e5 e5 e5 e5 e5 e5 e5 |etpwd0..........|
f94b77d0: 30 b4 18 7d 28 7f XX XX XX  XX XX XX XX XX XX XX |0..}(...........|
f94b77e0: XX XX XX XX XX XX XX XX XX  XX XX XX XX XX XX XX |................|
f94b77f0: XX XX XX XX XX XX XX XX XX  XX XX XX XX XX XX XX |................|
f94b7800: e5 e5 e5 e5 e5 e5 e5 e5 e5  e5 e5 e5 e5 e5 e5 e5 |................|
f94b7810: 68 74 74 70 73 3a 2f 2f 61  64 64 6f 6e 73 2e 63 |https://addons.c|
f94b7820: 64 6e 2e 6d 6f 7a 69 6c 6c  61 2e 6e 65 74 2f 75 |dn.mozilla.net/u|
f94b7830: 73 65 72 2d 6d 65 64 69 61  2f 61 64 64 6f 6e 5f |ser-media/addon_|
f94b7840: 69 63 6f 6e 73 2f 33 35 34  2f 33 35 34 33 39 39 |icons/354/354399|
f94b7850: 2d 36 34 2e 70 6e 67 3f 6d  6f 64 69 66 69 65 64 |-64.png?modified|
f94b7860: 3d 31 34 35 32 32 34 34 38  31 35 XX XX XX XX XX |=1452244815.....|
```


AND IN OTHER NEWS...
MELTDOWN
WE'RE ALL DOOMED, SANDRA.
HOW ABOUT THE WEATHER?
COMEDY CENTRAL

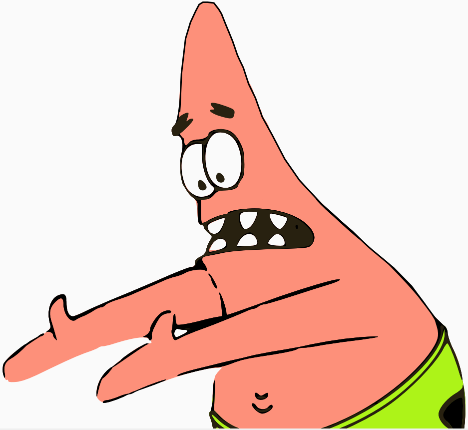
Not so fast. . .

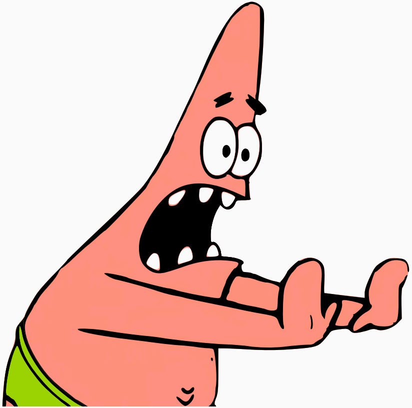
Take the kernel addresses...

- Kernel addresses in user space are a problem

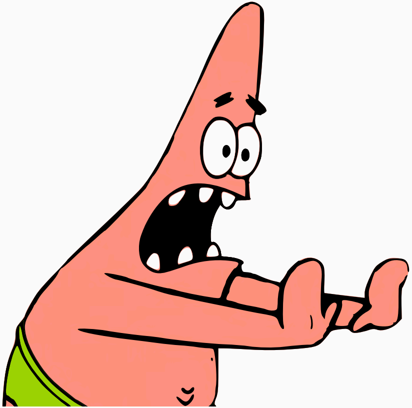
Take the kernel addresses...

- Kernel addresses in user space are a problem
- Why don't we take the kernel addresses...





- ...and remove them if not needed?



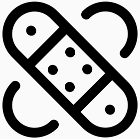
- ...and remove them if not needed?
- User accessible check in hardware is not reliable

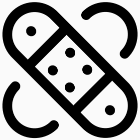
Kernel **A**ddress **I**solation to have **S**ide channels **E**fficiently **R**emoved

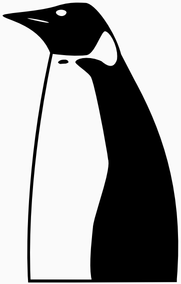
KAISER /ˈkʌɪzə/

1. [german] Emperor, ruler of an empire
2. largest penguin, emperor penguin

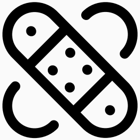
Kernel **A**ddress **I**solation to have **S**ide channels **E**fficiently **R**emoved







- Our patch
- Adopted in Linux



- Our patch
- Adopted in Linux

- Adopted in Windows

KAISER Patches

- Our patch
- Adopted in Linux

- Adopted in Windows

- Adopted in OSX/iOS

KAISER Patches

- Our patch
- Adopted in Linux

- Adopted in Windows

- Adopted in OSX/iOS

→ **now in every computer**


»A table for 6 please«

Speculative Cooking


»A table for 6 please«

PIZZA

SPECIAL RECIPES

PIZZA

SPECIAL RECIPES

Pizza


PIZZA
SPECIAL RECIPES


Pizza

PIZZA

SPECIAL RECIPES

Pizza

```
index = 0;

char* data = "textKEY";

if (index < 4)
```

then

Prediction

else

```
LUT[data[index] * 4096]
```

0

Spectre (variant 1)


```
index = 0;

char* data = "textKEY";

if (index < 4)
```

then

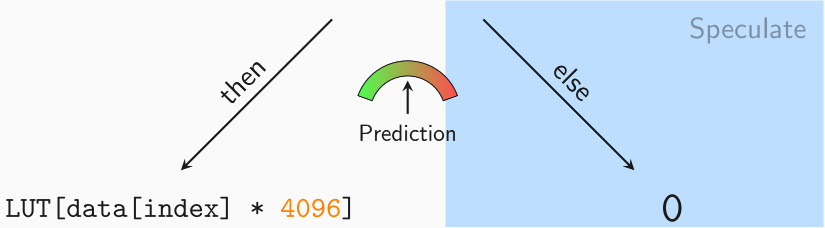
Prediction

else

```
LUT[data[index] * 4096]
```

0

```
index = 0;

char* data = "textKEY";

if (index < 4)
```

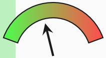


```
index = 0;

char* data = "textKEY";

if (index < 4)
```

Execute

then

```
LUT[data[index] * 4096]
```



Prediction

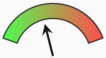
else

0

```
index = 1;

char* data = "textKEY";

if (index < 4)
```

then

else



Prediction

```
LUT[data[index] * 4096]
```

```
0
```

Spectre (variant 1)

```
index = 1;

char* data = "textKEY";

if (index < 4)
```

then

else

Prediction

```
LUT[data[index] * 4096]
```

0

```
index = 1;

char* data = "textKEY";

if (index < 4)
```

Speculate

then

else

Prediction

```
LUT[data[index] * 4096]
```

0

```
index = 1;

char* data = "textKEY";

if (index < 4)
```

then

else

Prediction

```
LUT[data[index] * 4096]
```

0

```
index = 2;

char* data = "textKEY";

if (index < 4)
```

then

Prediction

else

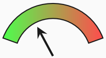
```
LUT[data[index] * 4096]
```

0

```
index = 2;

char* data = "textKEY";

if (index < 4)
```

then

else



Prediction

```
LUT[data[index] * 4096]
```

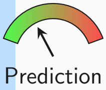
0

```
index = 2;

char* data = "textKEY";

if (index < 4)
```

Speculate

then

else




```
LUT[data[index] * 4096]
```

0

```
index = 2;

char* data = "textKEY";

if (index < 4)
```

then

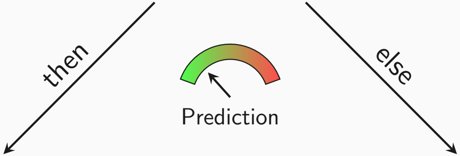
Prediction

else

```
LUT[data[index] * 4096]
```

0

```
index = 3;

char* data = "textKEY";

if (index < 4)
```




```
LUT[data[index] * 4096]                    0
```

```
index = 3;

char* data = "textKEY";

if (index < 4)
```

then

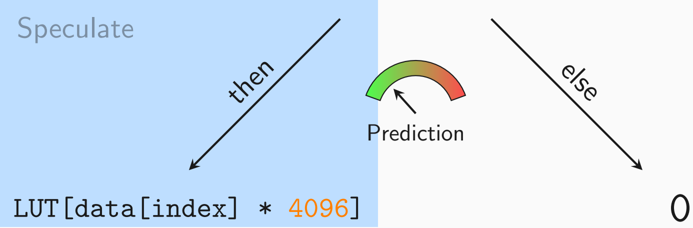
else

Prediction

```
LUT[data[index] * 4096]
```

0

```
index = 3;

char* data = "textKEY";

if (index < 4)
```



```
index = 3;

char* data = "textKEY";

if (index < 4)
```

then

Prediction

else

```
LUT[data[index] * 4096]
```

0

```
index = 4;

char* data = "textKEY";

if (index < 4)
```

then

else

Prediction

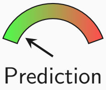
```
LUT[data[index] * 4096]
```

0

```
index = 4;

char* data = "textKEY";

if (index < 4)
```

then

else




```
LUT[data[index] * 4096]
```

0

```
index = 4;

char* data = "textKEY";

if (index < 4)
```

Speculate

then

Prediction

else

```
LUT[data[index] * 4096]
```

0

```
index = 4;

char* data = "textKEY";

if (index < 4)
```

then

Prediction

else

Execute

```
LUT[data[index] * 4096]
```

0

```
index = 5;

char* data = "textKEY";

if (index < 4)
```

then

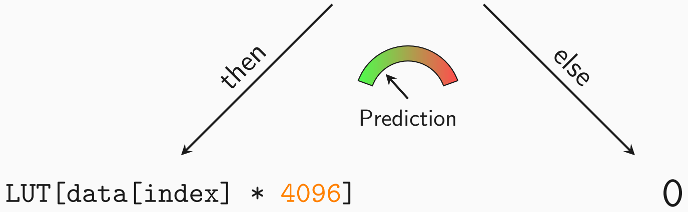
Prediction

else

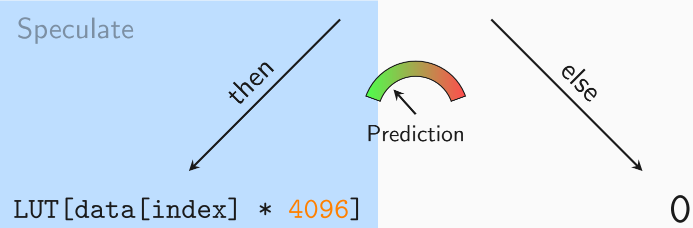
```
LUT[data[index] * 4096]
```

0

```
index = 5;

char* data = "textKEY";

if (index < 4)
```



```
index = 5;

char* data = "textKEY";

if (index < 4)
```



```
index = 5;

char* data = "textKEY";

if (index < 4)
```

then

Prediction

else

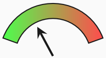
Execute

```
LUT[data[index] * 4096]
```

0

```
index = 6;

char* data = "textKEY";

if (index < 4)
```

then



Prediction

else

```
LUT[data[index] * 4096]
```

0

```
index = 6;

char* data = "textKEY";

if (index < 4)
```

then

Prediction

else

```
LUT[data[index] * 4096]
```

0

```
index = 6;

char* data = "textKEY";

if (index < 4)
```

Speculate

then

Prediction

else

```
LUT[data[index] * 4096]
```

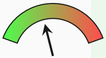
0

```
index = 6;

char* data = "textKEY";

if (index < 4)
```

then

else

Execute



Prediction

```
LUT[data[index] * 4096]
```

0


BLOCKCHAIN
HD
HISTORY.COM

ACM SIGARCH

JOIN BENEFIT CONTRIBUTE DISCOVER BLOG

Computer Architecture Today

Informing the broad computing community about current activities, advances and future directions in computer architecture.

Let's Keep it to Ourselves: Don't Disclose Vulnerabilities

by Gus Uht on Jan 31, 2019 | Tags: Opinion, Security

CONTRIBUTE

Editor: Alvin R. Lebeck

Associate Editor: Vijay Janapa Reddi

Contribute to Computer Architecture Today

What do we learn from it?

We have ignored microarchitectural attacks for many many years:

What do we learn from it?

We have ignored microarchitectural attacks for many many years:

- attacks on crypto

We have ignored microarchitectural attacks for many many years:

- attacks on crypto → “software should be fixed”

We have ignored microarchitectural attacks for many many years:

- attacks on crypto → "software should be fixed"
- attacks on ASLR

What do we learn from it?

We have ignored microarchitectural attacks for many many years:

- attacks on crypto → “software should be fixed”
- attacks on ASLR → “ASLR is broken anyway”

What do we learn from it?

We have ignored microarchitectural attacks for many many years:

- attacks on crypto → “software should be fixed”
- attacks on ASLR → “ASLR is broken anyway”
- attacks on SGX and TrustZone

What do we learn from it?

We have ignored microarchitectural attacks for many many years:

- attacks on crypto → "software should be fixed"
- attacks on ASLR → "ASLR is broken anyway"
- attacks on SGX and TrustZone → "not part of the threat model"

We have ignored microarchitectural attacks for many many years:

- attacks on crypto → “software should be fixed”
- attacks on ASLR → “ASLR is broken anyway”
- attacks on SGX and TrustZone → “not part of the threat model”
- Rowhammer attacks

What do we learn from it?

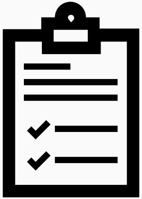
We have ignored microarchitectural attacks for many many years:

- attacks on crypto → "software should be fixed"
- attacks on ASLR → "ASLR is broken anyway"
- attacks on SGX and TrustZone → "not part of the threat model"
- Rowhammer attacks → "only affects cheap sub-standard modules"

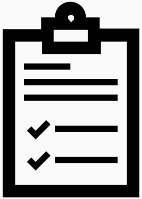
We have ignored microarchitectural attacks for many many years:

- attacks on crypto → "software should be fixed"
- attacks on ASLR → "ASLR is broken anyway"
- attacks on SGX and TrustZone → "not part of the threat model"
- Rowhammer attacks → "only affects cheap sub-standard modules"

→ for years we solely optimized for performance

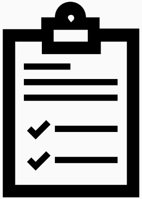


After learning about a side channel you realize:



After learning about a side channel you realize:

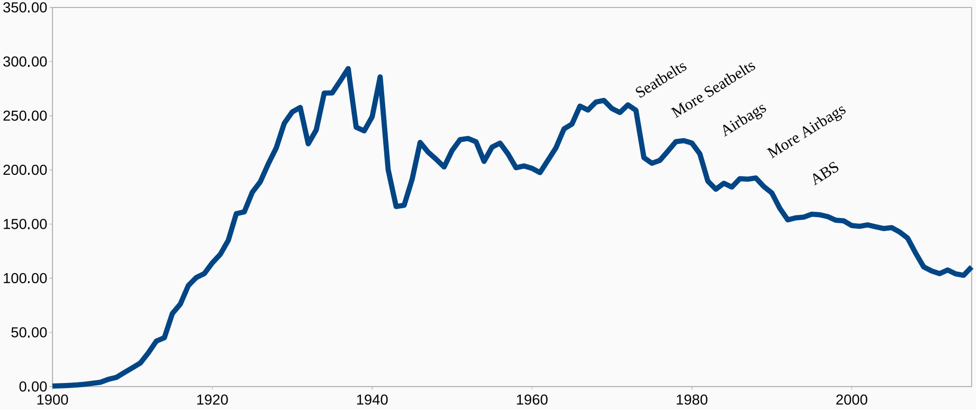
- the side channels were documented in the Intel manual



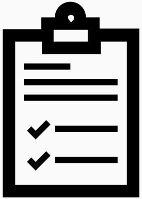
After learning about a side channel you realize:

- the side channels were documented in the Intel manual
- only now we understand the implications

What do we learn from it?



Motor Vehicle Deaths per Year (normalized by US population)



A unique chance to

- for the security industry and academia to grow up
- find good trade-offs between security and performance, efficiency, and complexity
- like the health sector, learn to cope with diseases

Software-based Microarchitectural Attacks: What do we learn from Meltdown and Spectre?

Daniel Gruss

March 26, 2019

Graz University of Technology