CASPER

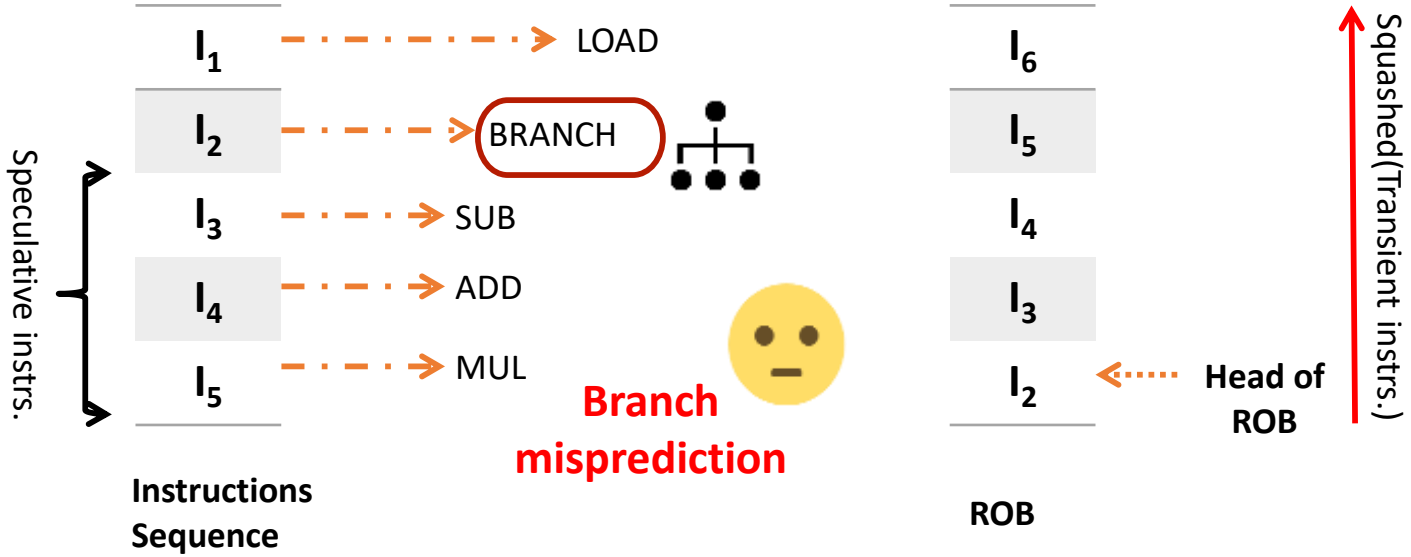
CS773-2025-Spring: Computer Architecture for Performance and Security

Lecture 9: Spectre Variant-II and Meltdown

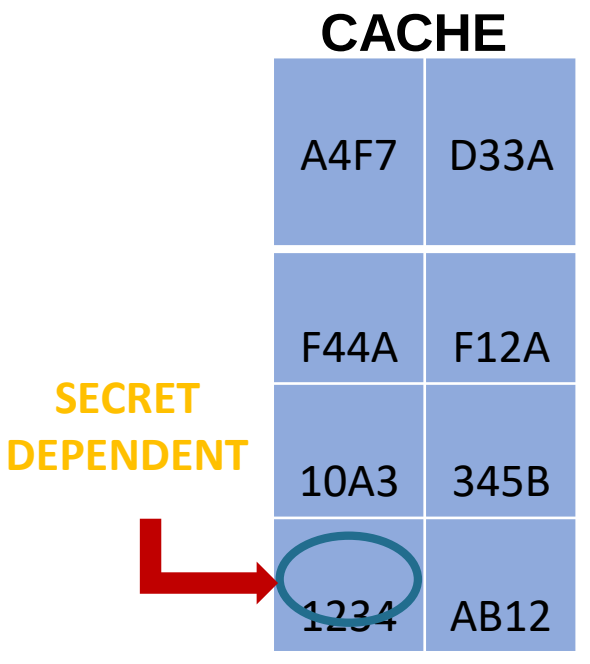
https://www.cse.iitb.ac.in/~biswa/


ON SILENT MODE PLEASE

The threat model [Transient execution attacks]



But, microarchitectural traces remain



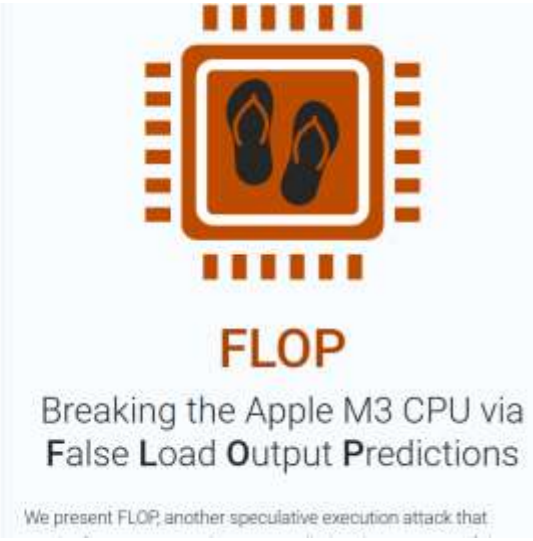
Spectre: More details

- Variant –I: The attacker tries to affect the prediction of branch outcome. However, the branch predictor is indexed by PC. In reality, some bits of PC. For example, a 1024-entry branch predictor, then a 64-bit PC is hashed into a 10-bit index for the branch predictor.
- Spectre (variant-I) can happen between two threads of the same core.
- So it works even on a single core. Can happen at L1 cache/L2 cache/L3 cache
- It is also possible to attack a core that is not using SMT. How? Context switch: The attacker comes, mis-trains, and a context switch happens, and the victim comes and starts accessing illegitimate addresses.
- Variant-II: It uses Branch Target Buffer and not branch predictor. Attacker writes an instruction that puts target into an address that is not accessible. This is also known as *branch target injection. Variant-I was bound check bypass.* Indirect `JMP` instructions consult the indirect branch predictor to direct speculative execution to the most likely target of the branch.

- Variant-II continues

An attacker can jump to a spectre gadget code by mistraining the BTB so that a victim will jump to that code snippet and start executing it speculatively.

Last 7 years: Spectre continues to ….



Summary of Spectre Attack: As an attacker

"Two paths diverged in a branch, and I—
I took the one less traveled by,
And that has made all the difference.“ ☺

-Frobert Rost, *The Path/Road Not Taken*

"Two paths diverged in a branch, and I—
I took the one more traveled by,
And that has made all the difference.“ ☺

-Frobert Rost, *The Path Taken*

Please read the paper (only one Section)

- Spectre-Variant (I and II)

Meltdown: The O3 Curse!!

1. raise_exception();
2. // line below is never reached
3. secret=KernelArray[data*4096];

Kernel Trap

Out-of-order (O3) as
it has no dependency

1. secret=KernelArray[data*4096];
2. raise_exception();

What about page-fault?

https://meltdownattack.com/meltdown.pdf

How does it work? Same way as Spectre

- Based on secret value, you access a particular cache set. The beauty of a meltdown attack is that you can dump the entire kernel into the cache ☹
- Mitigation: Let's understand the problem first. Before the Meltdown attack, the CR3 register that stores the base address of the page table remained the same, even if the user switched to kernel mode.

What next: All addresses of the kernel is mapped to all processes too ☺

Why? Switching time overhead is high if it wont be mapped
Solution: KPTI (kernel page table isolation). A new CR3 register only for the kernel page table. So a user cannot access kernel addresses in user mode even speculatively ☺ Applied successfully on several Intel processors on various OSes (Linux-2.6.32 to 4.13.0), Windows 10, Docker, LXC, and OpenVZ.

Real World Real Melt and Down

```
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.....|
```

Listing 4: Memory dump of Firefox 56 on Ubuntu 16.10 on a Intel Core i7-6700K disclosing saved passwords (cf.

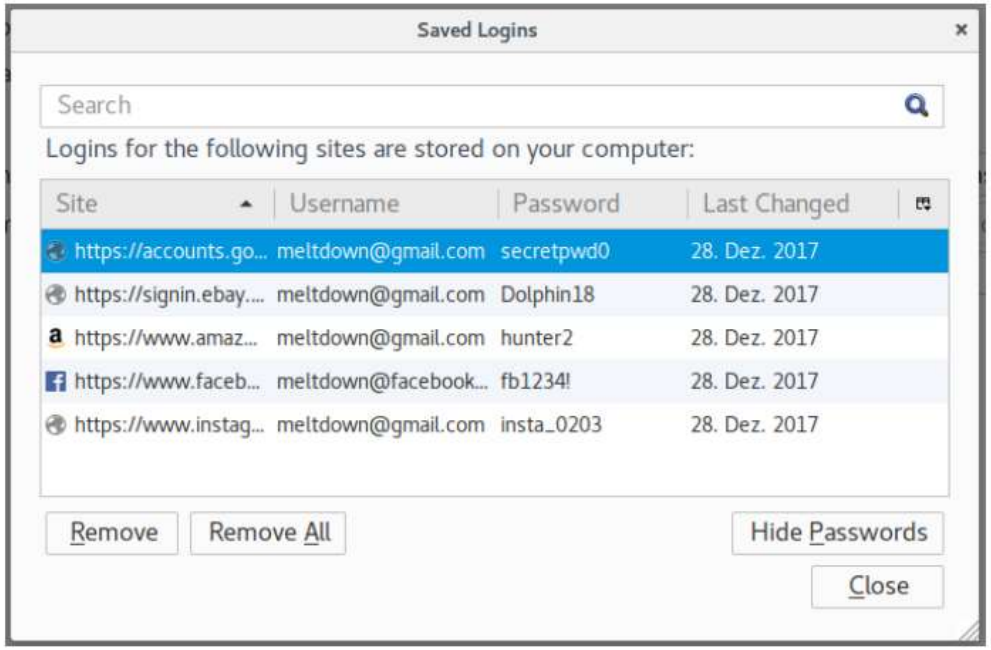


Figure 6: Firefox 56 password manager showing the stored passwords that are leaked using Meltdown in Listing 4.

Readings

- Spectre and Meltdown: https://meltdownattack.com/
- For meltdown, make sure KASLR is disabled if you want to attack now.
- ASLR (Address space layout randomization) randomizes the location of binaries in memory.
- Also make sure, you disable meltdown mitigation patch