CASPER

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

Lecture 8: Transient Execution Attacks ☺

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


ON SILENT MODE PLEASE

Spectre and Meltdown

Spectre in Action: Fasten Your Seat Belts

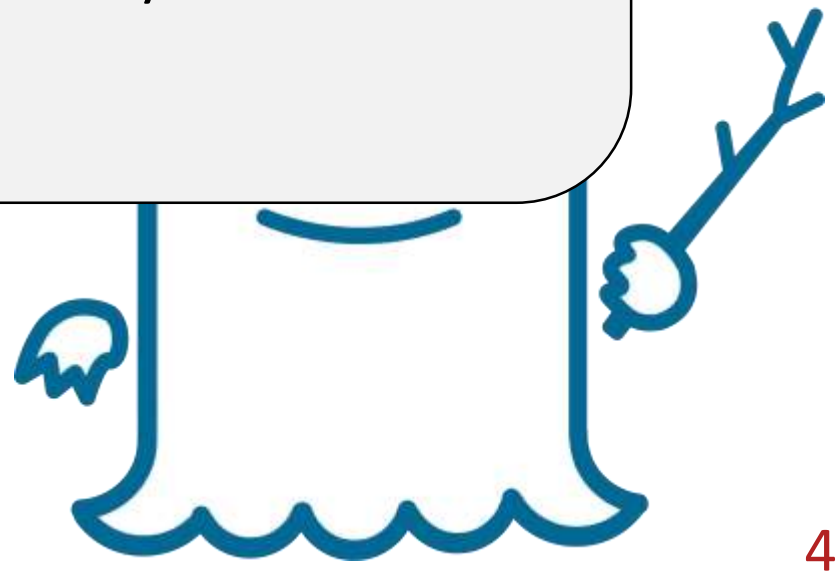
```
int CS773Array = [100, 200, 300];
int attacker = 4;
if (attacker < sizeof(CS773Array))
        x = CS773Array[attacker]

y=MyArray[CS773Array[attacker]*512]
```

DRAM LOAD (make sure you thrash all from cache)

DRAM LOAD (make sure you thrash all from cache)

X is the secret that is not accessible to the attacker



Branch Predictor and Speculative Execution

```
int CS773Array = [100, 200, 300];
int attacker = 4;
if (attacker < sizeof(CS773Array))
        y = MyArray[CS773Array[attacker]*512]
```

Branch predictor returns TRUE ☹

Branch Predictor and Speculative Execution

```
int CS773Array = [100, 200, 300];
int attacker = 4;
if (attacker < sizeof(CS773Array))
        y = MyArray[CS773Array[attacker]*512]
```

Branch predictor returns TRUE ☹

T T T T T T T T T T T T

Attacker has mis-trained it ☹ ☹

How? By using values less than 3 always ☹ ☹

Branch Predictor and Speculative Execution

```
int CS773Array = [100, 200, 300];
int attacker = 4;
if (attacker < sizeof(CS773Array))
        y = MyArray[CS773Array[attacker]*512]
```

Branch predictor returns TRUE ☹

Attacker has mis-trained it ☹ ☹

Processor is on the wrong-path ☹ ☹ ☹

Branch Predictor and Speculative Execution

```
int CS773Array = [100, 200, 300];
int attacker = 4;
if (attacker < sizeof(CS773Array))
        y = MyArray[CS773Array[attacker]*512]
```

Branch predictor returns TRUE ☹

Attacker has mis-trained it ☹ ☹

Processor is on the wrong-path ☹ ☹ ☹

Branch resolution latency 200 cycles ☹ ☹ ☹ ☹

Within these 200 cycles ☺

```
int CS773Array = [100, 200, 300];
int attacker = 4;
if (attacker < sizeof(CS773Array))
        y = MyArray[CS773Array[attacker]*512]
```

CS773Array[4] is in L1/L2/L3 ☹

The address is in the cache ☹ ☹

Yes, you guessed it right: F+R, P+P cache attacks ☹ ☹ ☹

After say 200 cycles

Processor realized it was a mistake and *squashed* all wrong path instructions

But cache has the data ☹

y = MyArray[CS773Array[attacker]*512]

LOAD MyArray[0] 60 ns
LOAD MyArray[512] 60 ns
LOAD MyArray[1024] 5 ns Bingo !! *CS773Array[attacker] = 2*

What if myarray is not shared with the attacker?

- Still it is possible to do Spectre attack ☺
- How?
- CS773Array[attacker] = 0, you access set 0
- CS773Array[attacker] = 1, you access set 1
- CS773Array[attacker] = k, you access set k

So, in summary, the attacker can do prime and probe on sets and check which sets show latency difference