The Spectre Vulnerability

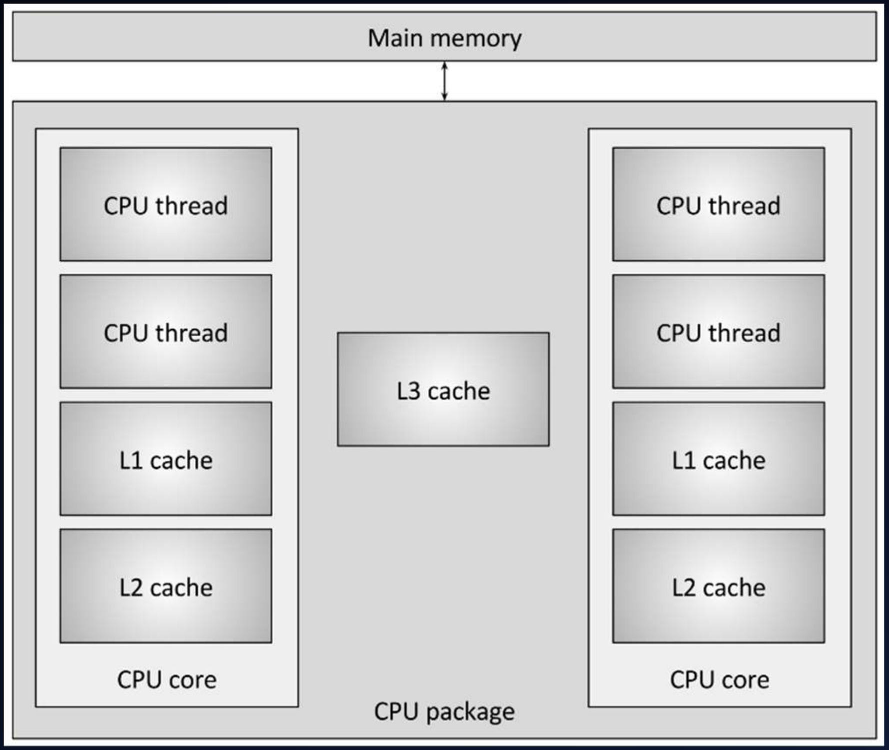
INVISIBLE BUGS

Several Key Concepts

- CPU Memory Caching
- Branch Prediction + Speculative Execution
- Side Channel Attacks
  - Timing Attacks

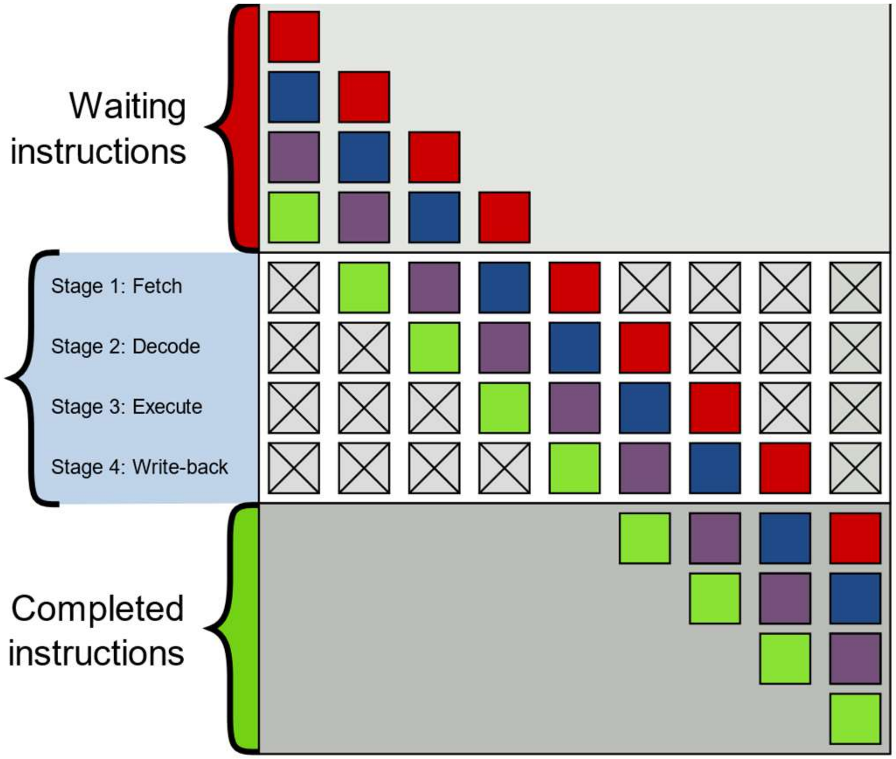
CPU Memory Caching

- Levels of Caching
  - L1, L2, and L3 cache is typically used in modern architectures
- **Static** RAM (vs. Dynamic RAM)
- Before going to DRAM, L1, L2, L3 is checked first
  - *Cache Miss* incurred if not in Cache, which takes a lot of time



Branch Prediction and Speculative Execution

- *Branch Prediction* is used in pipelined systems to prevent bottlenecking due to DRAM retrievals
- CPU trained to predict likely path of branching statement (can be wrong)
  - AMD uses AI neural network
  - BTB components in a Branch Prediction Unit map historical jumps
- Code inside predicted branch is *speculatively executed*, and results are cached



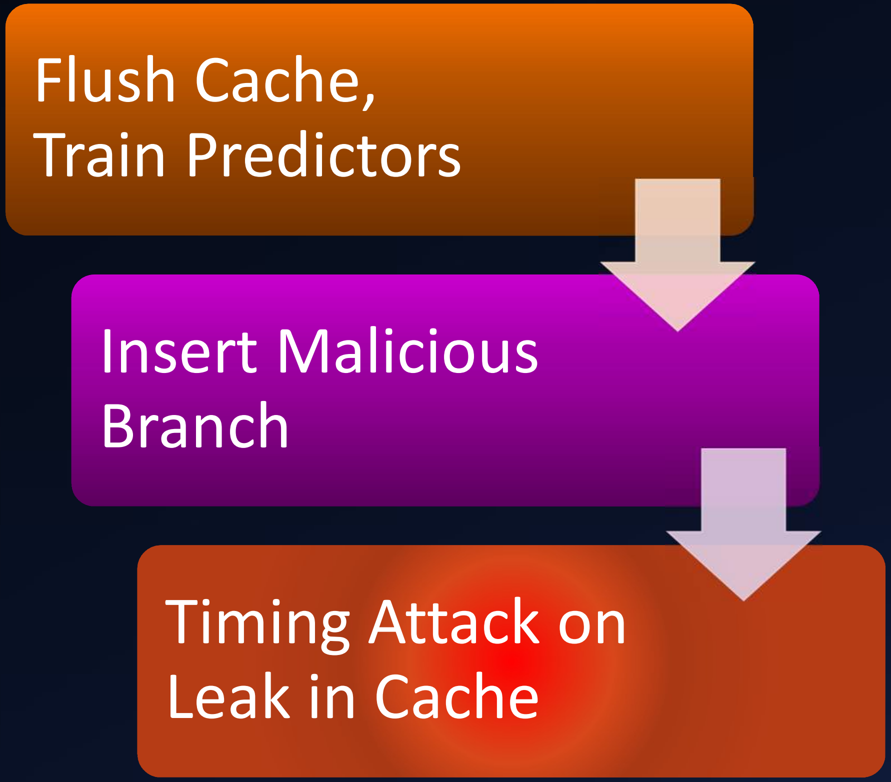
Side Channel Attacks

Exploiting "side effects" to *infer data* instead of directly accessing data

- Timing Attack
  - Uses the "side effect" of *time* spent on a computation in order to infer critical data or information
  - Anomalies in computation time can provide side channel into sensitive data.

How Does Spectre Work?

- Branch Predictors trained to expect a specific branch to be true (with valid code)
- Malicious Branch + Code inserted
- Timing Attack on Cache after Speculative Execution
  - *Processor does not know whether the speculatively executed code is illegal (happens before exception is thrown)*

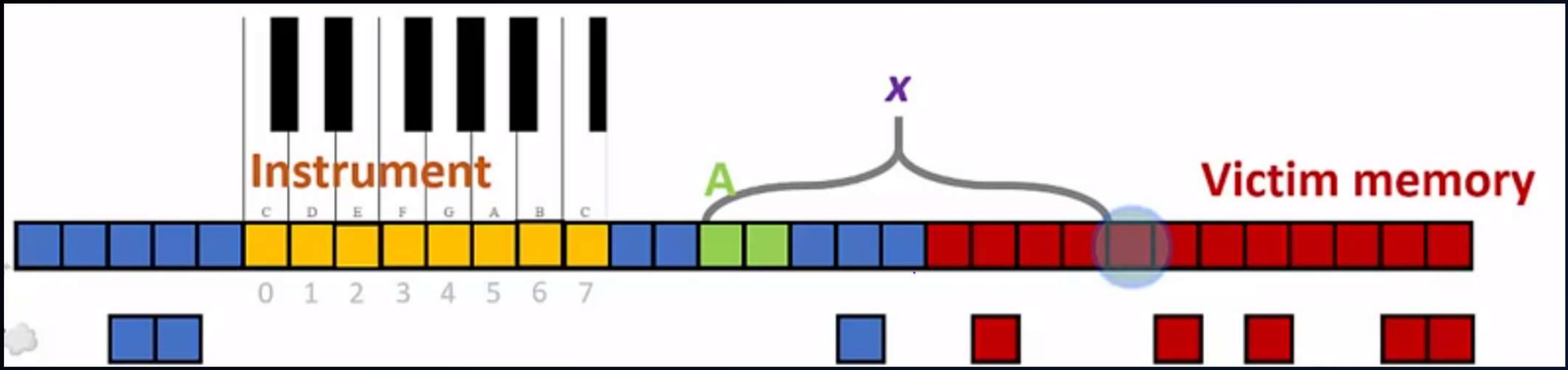


Flush Cache,
Train Predictors

Insert Malicious
Branch

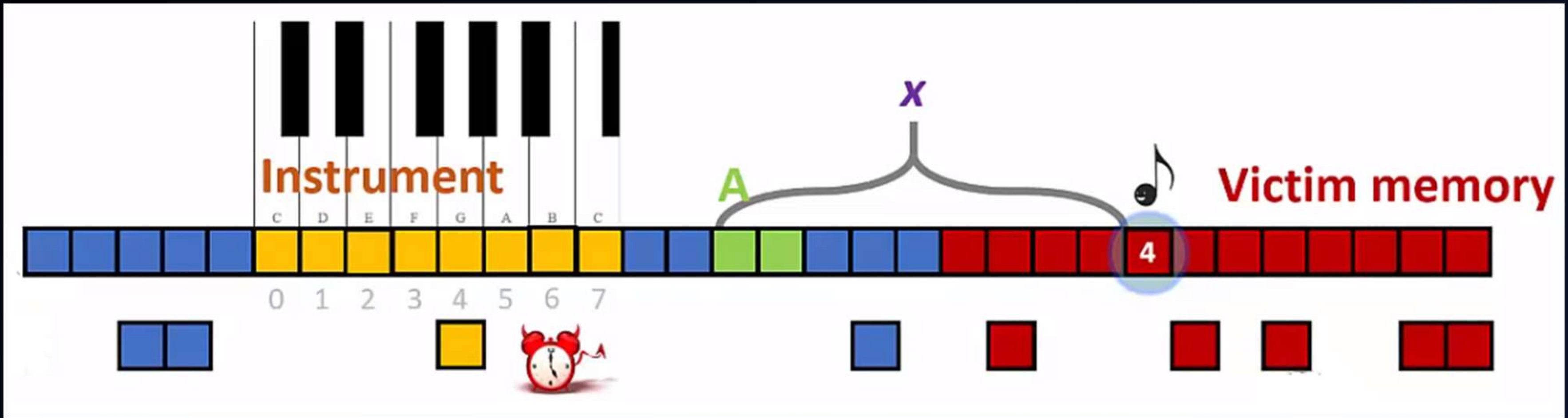
Timing Attack on
Leak in Cache

Spectre In Action

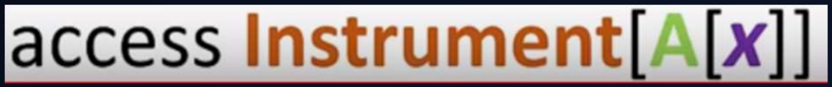


Cached Memory

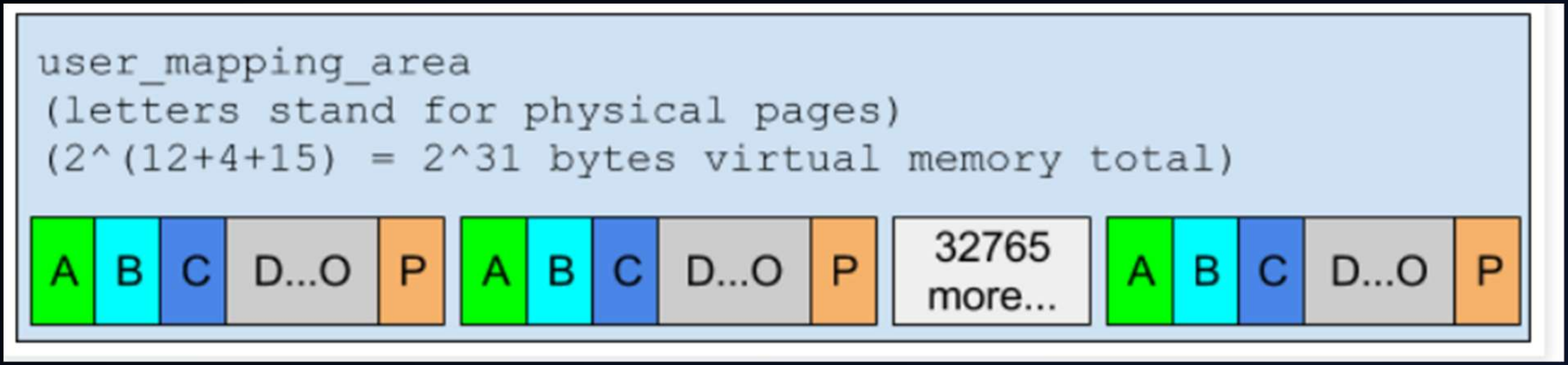
Spectre In Action



Inside Malicious Branch, Speculatively Execute


access Instrument[A[x]]

Actual Instrument Character Mapping is Slightly More Sophisticated…



```
user_mapping_area
(letters stand for physical pages)
(2^(12+4+15) = 2^31 bytes virtual memory total)
```

https://googleprojectzero.blogspot.com/2018/01/reading-privileged-memory-with-side.html

<end>

INVISIBLE BUGS

Image Credits

- https://meltdownattack.com/ - Spectre Main Logo
- https://techviral.net/hackers-can-access-your-calls-messages-by-using/ - Masked Hacker
- https://www.youtube.com/watch?v=mgAN4w7LH2o – Spectre Instrument Demonstration Captures
- https://www.youtube.com/watch?v=yi0FhRqDJfo – CPU Cache Animation
- https://googleprojectzero.blogspot.com/2018/01/reading-privileged-memory-with-side.html - Actual Implementation of Mapping Array
- By en:User:Cburnett - Own work This vector image was created with Inkscape., CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=1499754 – 4 Stage Pipelining