Plan
- Recitation Qs
- Meltdown
  - Load kernel data into cache.
  - Read kernel data from cache.

Logistics
- Design project due May 11
- If you like this, consider 6.3060 in Fall 2021
Recitation Questions

1. What is the meltdown attack?
   - Use 000 execution to read "privileged data" from unprivileged process.
   - Side-channel attack... exploiting unspecified behavior

2. How does it work?
   - "Transient instruction sequence"

3. Why is this attack possible?
   - CPU designers prioritize speed
**Meltdown**

**Goal:** Read data of another user on the same machine.
- Email
- Password
- Crypto key

This attack will no longer work.

→ "Spectre" attacks.
**Meltdown (restated):** Read arbitrary addr in memory, bypassing HW permissions.

Virtual Memory

Physical Memory

16 GB
Step 1: Load kernel data into register.

```c
int main() {
    char buf[256*128];
    byte k = *kernel_addr;
    char stuff = buf[k*128];
}
```

CPU will...
- Load data from memory
- Cache the data
- Check permissions
- Crash process if permission fails
- Execute next instruction (speculative execution)
Possible pros to avoid crashing:

\[ \text{char } k = \begin{cases} 
\text{buf}(k) & \text{FAST} \\
\text{buf}(0) & \text{SLOW} \\
\text{buf}(2) & \\
\text{buf}(3) & \\
\text{buf}(255) & 
\end{cases} \]

\[ \Rightarrow \text{256 possible vals of } k \]

... try them all

Step 3: Figure out which element of buf the CPU accessed before crashing.
Prefetching

but [3], ---, but [3-128]
Mitigations

OS/WS

Kernal data

User data

HW

protected by HW Penn (i.e.)

HW
Plan
- Recitation Qs
- Meltdown
  - Load kernel data into cache.
  - Read kernel data from cache.

Logistics
* Design project due May 11
* If you like this, consider 6.5060 in Fall 2021
1. What is Meltdown?
   - Attack that uses zero execution

2. How does it work?
   - Read cached results... covert channel

3. Why is the attack possible?
   - Perf benefits
   - Race condition
Meltdown

Goal: Read data of another user
- email, password, crypto key, ...

Machine

This attack no longer works

by "Spectre attack"
Meltdown

not allowed by HW

Meltdown

Virtual addr

Physical Mem

0x172

0x353

16 GB

0

KAISER / KPTI
Step 1: Load kernel data into register.

```c
int main() {  
  byte buf[256];  
  byte k = *kernel_addr;  
  byte stuff = buf[k];  
  CPU will...  
  - Load data from num  
  - Check perm bits  
  - Crash prog if perm check fails; say "no"  
  - Execute next instruction  
  making coffee
```
Execution engine

byte k =
buf[k]
buf[0]  

Cache

k

RAM (memory)
16 GB

buf

buf(0)
buf(1)
buf(256)

clflush
byte buf[128×256]

"Prefetching"
Mitigation

HW:
- Don't speculate at all. 😞
- Separate caches
- Don't speculate past permission checks (?)

Software: