Meltdown and Spectre: Complexity and the death of security

Dr. Michael S. Kirkpatrick
May 8, 2018
Meltdown and Spectre: 
Wait, my computer does what?

Dr. Michael S. Kirkpatrick
May 8, 2018
Meltdown and Spectre:
Whoever thought that was a good idea?

Dr. Michael S. Kirkpatrick
May 8, 2018
Meltdown and Spectre:
I give up. Can I just retire now?

Dr. Michael S. Kirkpatrick
May 8, 2018
No one alive understands how computers behave.
Computers are complex systems.
Read kernel location deadbeef
Memory hierarchy and cache

1: Access allowed?
2: OK, here’s your data

1: Here’s your data
2: Access allowed?
x86 Pipelining

If orange 5 doesn’t depend on 3 & 4, why wait?
x86 Pipelining

1 2 1 1 2 2
3 4 5 4 5 5 3 3 4
6 7 7 6 6 8 8 7 8
What we know so far…

- You’re not supposed to access kernel
- L1 cache timing is wrong
- x86 pipelining is complex
- Macroarchitecture != microarchitecture
  - “First” != First
- “Invisible” side effects are visible
Cache-based side channels

Guess what q is!

\[ x = \text{array}[q]; \]

for (i = 0; i < 8; i++) {
    start_timer();
    y = \text{array}[i];
    stop_timer();
}
Meltdown

Meltdown consists of 3 steps:

1. **Step 1: Reading the secret.**
   - `rcx = kernel address`
   - `rbx = probe array`
   - `mov rbx, qword [rcx]`
   - Multiply byte by 4096
   - Use value to hit cache line

2. **Step 2**

3. **Step 3**

4. **Step 4**

8: Maybe I should check if step 4 was valid…

Listing 2: The core instruction sequence of Meltdown.

```
1 ; rcx = kernel address
2 ; rbx = probe array
3 ret
4 mov rbx, qword [rcx]
5 shl
6 jz
7 mov rbx, qword [rbx + rax]
8: Maybe I should check if step 4 was valid…
```

Read a byte of the kernel

Multiply byte by 4096

Use value to hit cache line
Meltdown

for (i = 0; i < 8; i++) {
    start_timer();
    y = array[i];
    x = array[y];
    stop_timer();
}
Meltdown

Physical memory

User

Kernel

Process memory contains...

the kernel, which contains…

physical memory, which contains…

the memory contents of every process.
Meltdown

Short-term fix
KAISER/PTI/KVAS

Long-term fix
Split address space
Replace hardware

kernel
stack
heap
data
code
kernel
stack
heap
data
code
kernel
x86 Pipelining
Speculative execution

```c
if (x < array_length)
    y = array[x];
```

- **x** is 2 → **y** becomes **c**
- **x** is 5 → **y** becomes **f**
- **x** is 1 → **y** becomes **b**
- **x** is 327 → **y** becomes
Spectre variant 1

\[
\text{if (} x < \text{array1\_size} \text{)} \\
y = \text{array2[array1[x]]} * 256;
\]

\( x = &\text{target} - &\text{array1} \) is the target

Cache hit

Cache miss

Cache miss

Cache hit

Cache hit

\( x = \&\text{target} - \&\text{array1} \) is the target
Spectre variant 2 widgets

Widget Count vs Number of Instructions

- Instructions per Widget
- Number of Widgets of that Size
- Windows 10
- Ubuntu 16.04
- OSX 10.13

https://34c3.cyber-itl.org/slides.pdf
<table>
<thead>
<tr>
<th></th>
<th>Meltdown</th>
<th>Spectre</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Short-term fix</strong></td>
<td>KAI SER/PTI/ KVAS</td>
<td>Microcode patch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OS update</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Recompile binaries</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Change compiler</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Browser hardening</td>
</tr>
<tr>
<td><strong>Long-term fix</strong></td>
<td>Split address space</td>
<td>????</td>
</tr>
<tr>
<td></td>
<td>Replace hardware</td>
<td></td>
</tr>
</tbody>
</table>

What about applications?
# CVE-2018-5093: Buffer overflow in WebAssembly during Memory/Table resizing

<table>
<thead>
<tr>
<th>REPORTER</th>
<th>OSS-Fuzz</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPACT</td>
<td>HIGH</td>
</tr>
</tbody>
</table>

**Description**
A heap buffer overflow vulnerability may occur in WebAssembly during Memory/Table resizing, resulting in a potentially exploitable crash.

**References**
- Bug 1415291

# CVE-2018-5094: Buffer overflow in WebAssembly with garbage collection on uninitialized memory

<table>
<thead>
<tr>
<th>REPORTER</th>
<th>OSS-Fuzz</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMPACT</td>
<td>HIGH</td>
</tr>
</tbody>
</table>
And so it begins…

Skyfall and Solace

More vulnerabilities in modern computers.

Following the recent release of the Meltdown and Spectre vulnerabilities, CVE-2017-5175, CVE-2017-5753 and CVE-2017-5754, there has been considerable speculation as to whether all the issues described can be fully mitigated.

Skyfall and Solace are two speculative attacks based on the work highlighted by Meltdown and Spectre.

Full details are still under embargo and will be published soon when chip manufacturers and Operating System vendors have prepared patches.

Watch this space…
And so it begins...

Hackers can bypass Windows Meltdown patch, and early builds may be at risk

Microsoft's Spectre/Meltdown patches for Windows 10 could be completely bypassed, and only users with the April 2018 Update are protected.

By Brandon Vigliarolo

Eight new Spectre variants affecting Intel chips discovered, four are "high risk"

Intel is already working on fixes

By Rob Thubron
A return to the past...
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 ROUHAMMER.

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.
Thank you and good luck!