MELTDOWN AND SPECTRE

Mile-high View

- Generally speaking, processes aren’t supposed to read data in memory that is being used by other processes.
- Almost all modern processors have design flaws that make this possible.
- Passwords copied from password managers, pictures, sensitive documents, PII, etc. can be stolen.
- Vulnerabilities go by names like KAISER, KPTI, F***KWIT.
- Enter “Meltdown” and “Spectre”.

Three for the price of one!

Exploited Vulnerability	CVE	Exploit Name	Public Vulnerability Name	Windows Changes	Silicon Microcode Update ALSO Required on Host
Spectre	2017-5753	Variant 1	Bounds Check Bypass	Compiler change; recompiled binaries now part of Windows Updates Edge & IE11 hardened to prevent exploit from JavaScript	No
Spectre	2017-5715	Variant 2	Branch Target Injection	Calling new CPU instructions to eliminate branch speculation in risky situations	Yes
Meltdown	2017-5754	Variant 3	Rogue Data Cache Load	Isolate kernel and user mode page tables	No

https://cloudblogs.microsoft.com/microsoftsecure/2018/01/09/understanding-the-performance-impact-of-spectre-and-meltdown-mitigations-on-windows-systems/

OWASP
Open Web Application Security Project

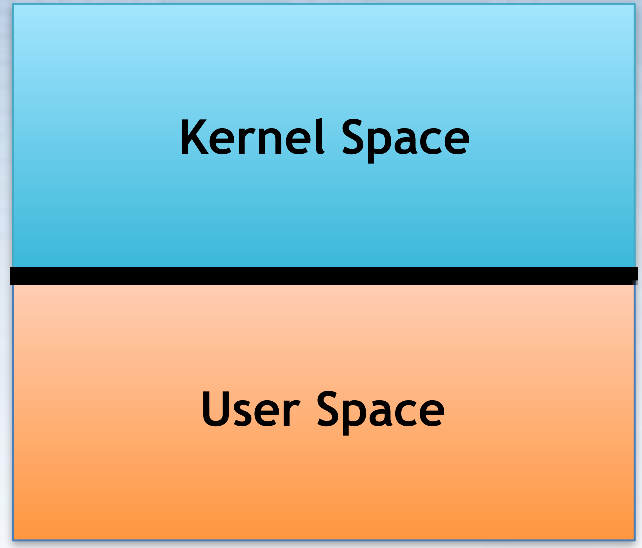
MELTDOWN

Meltdown

- Breaks (or “melts”) the fundamental barrier between user space (userland) and kernel space.

- Allows users to directly access the memory of other processes and the host OS.

- So far, it seems to affect Intel only.



Meltdown - How Does it Work?

- Processors utilize “out-of-order” execution of instructions.
- Important performance feature in modern processors.
- Processor starts working ahead on “likely future” tasks in a process while it is waiting on an earlier task in a process to complete.
- Like baking a cake.
- While the baker is monitoring the cake baking in the oven, he makes the frosting and puts it in a “cache” bowl, rather than waiting for the bake “process” to complete first.

Meltdown - How Does it Work?

- If the cake is baked properly, the baker applies the already made frosting. Performance win!
- If the cake is burned, the baker throws away the frosting and starts over (he’s finicky that way).
- The processor is the same. If the earlier tasks complete successfully, the later tasks are already completed, thus saving time.
- If the earlier task fails, the processor dumps the completed work and starts over. Lost time, but happens rarely.

OWASP
Open Web Application Security Project

Meltdown - How Does it Work?

- The problem? While the baker is making the cake, a thief reaches in the window and steals some frosting from the bowl.
- Affected CPUs allow unprivileged processes to load data from a privileged memory into a temporary CPU register where anyone can get it.
- An attacker can run a script to dump the entire kernel memory and export it to the outside world via a covert channel.
- https://meltdownattack.com/meltdown.pdf

Mr. Frosting Face

So wudda we do ‘bout it, huh?

- Kernel Address Isolation to have Side-channels Efficiently Removed (KAISER)
- Now called Kernel Page-Table Isolation (KPTI)
- Separates user-space and kernel-space page tables entirely
- 5%-30% performance hit? Virtualization could be hit hard.
- AV Update/Registry Key
- Implement in TEST first.

OWASP
Open Web Application Security Project

So wudda we do ‘bout it, huh?

- AV Update/Set Registry Key

  RegKey=“HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion_
  QualityCompat"

  Value Name ="cadca5fe-87d3-4b96-b7fb-a231484277cc"

  Type="REG_DWORD"

  Data=“0x00000000"

- Microsoft won’t allow updates until third party AV is updated and/or this registry key is set?

https://kc.mcafee.com/corporate/index?page=content&id=KB90167

SPECTRE

Spectre

- It will “haunt” us for years to come. Get it?
- Breaks the isolation between applications.
- No programming errors needed to exploit.
- Applications that are bug free and follow security best practices are vulnerable.
- Safety checks of said best practices might actually make applications more vulnerable

Irony can be pretty ironic sometimes.

Spectre

- Harder to exploit than Meltdown. Yay!
- Harder to mitigate, too. Ugh.
- Can be patched without a performance hit, **if** the exploit is **known.**
- The usual stuff could be lost. Passwords, financial data, pictures, etc.
- Intel, AMD, and most ARM processors affected.
- KAISER patch is of no help here.

Spectre - How does it work?

- Attacker injects a malicious instruction sequence in process address space - through a bug or not… it depends
- Attacker tricks the CPU into speculatively executing the malicious sequence
- Establishes a covert channel
- Memory and register contents are leaked across
- Attacker does the happy dance
- https://spectreattack.com/spectre.pdf

Happy Dancing Hacker

So wudda we do ‘bout it, huh?

- Update AV, MicroOS and browser software, firmware
  - SharedArrayBuffer will likely be disabled in most browsers until this is resolved.
  - Edge, FireFox, and Chrome will disable it in next release
  - Most other major browsers will follow suit.
  - If you are using a browser that still supports SharedArrayBuffer, upgrade or dump it.
- Firmware update could be the most difficult

So wudda we do ‘bout it, huh?

- Google Suggestions for Developers
    - Where possible, prevent cookies from entering the renderer process' memory by using the SameSite and HTTPOnly cookie attributes
    - Avoid reading from document.cookie.
    - Don’t serve user-specific or sensitive content from URLs that attackers can predict or easily learn. (e.g. <img class="lazy" data-src=”https://email.example.com/inbox.json”/>)
    - Use anti-CSRF tokens and SameSite cookies, or random URLs to mitigate this kind of attack.
    - Make sure your MIME types are correct
    - Specify a nosniff header for any URLs with user-specific or sensitive content
    - https://www.chromium.org/Home/chromium-security/ssca

QUESTIONS?

paul.kern@owasp.org