Introducing Meltdown & Spectre

The Headache of 2018

Andre van Eyssen, Chris Knowling

January 14, 2018.
Updated January 19, 2018.

Objective

This pack is intended to provide an overview of the recent documented exploitation of modern CPU (mis)features.

It is not designed to be a comprehensive document on the fine details of the exploits and specific platforms. Where possible, links will be provided to full documentation and disclosure.

Following is a description of the situation, remediation options and impacts, as well suggestions for remediation strategy.

Status

Information is rolling out daily on these subjects. This document is accurate as of the 14th of January.

And so it begins...

- Meltdown
  "technique can enable a user process to read kernel memory"
- Spectre
  "exploits out-of-order execution to leak the target's physical memory"

Meltdown

- Variant 3: rogue data cache load CVE-2017-5754
- Variant 3a: per above but for ARM Cortex A15, A57 and A72 only.

- https://meltdownattack.com/meltdown.pdf

Spectre

- Variant 1: bounds check bypass CVE-2017-5753
- Variant 2: branch target injection CVE-2017-5715

- https://spectreattack.com/spectre.pdf

What's impacted? (CPU)

- Meltdown
  - Intel: Every x86/x64 CPU since the Pentium-II (Klamath) core.[1]
  - ARM: Some Cortex ARM platforms.[2]
  - AMD: Vendor states no impact.[3]
- Spectre
  - Intel, AMD, ARM.
  - IBM POWER[4]
  - SPARCv9[5]
  - HP PA-RISC, large MIPS (eg, R10K)

---

[1] https://security-center.intel.com/advisory.aspx?intelid=INTEL-SA-00088&languageid=en-fr

[2] https://developer.arm.com/support/security-update

[3] http://www.amd.com/en/corporate/speculative-execution

[4] (per IBM)

[5] (per Oracle Jan 2018 Crit Disclosure)

What's impacted? (General)

- Servers
- Virtual Farm hosts – VMware, Xen, KVM, etc.
- Desktops
- Some mobile devices
- Storage arrays and Network elements
  - Especially x86 based like NetApp, F5, etc.
- Software "appliances"
  - These are usually just x86 Linux servers.
- Pretty much anything with a decent-size CPU.

Spectre – v1

- Allows malicious code to circumvent bounds checking features built into most binaries.
- The bounds checks still fail however the CPU will speculatively execute instructions after the bounds checks.
- Speculative code can access memory that would normally be prevented by the bounds check.
- The CPU discards instructions executed speculatively; however, system state changes can be observed, in particular CPU caches.
- Malicious code can detect these changes and read the data that was speculatively accessed.
- **It is difficult for a system to run untrusted code within a process and restrict which memory that code can access.**

Spectre – v2

- Processes can influence the speculative execution of code in other contexts[6] on the same physical core.
- CPUs attempt to predict destination for indirect jumps and calls and start speculatively executing at the predicted location. Prediction tables are shared between processes on the same core.
- A process can pollute the prediction tables to influence the prediction of other code.
- A process can cause speculative execution of any mapped code in another context.
- **An attack could read memory from another virtual guest, process or the kernel similiar to variant 1.**

[6] CPU ring, process, guest/host mode

Meltdown – v3/v3a

- Allows malicious user mode code to access virtual memory as if the process was executing in a kernel context.
- On some processors, the speculative execution of code can access any memory on the host as if it was the kernel VM system.
- Similar to variant 1, a process can read memory that was accessed speculatively. However variant 3 potentially allows access to anything the kernel can access – other processes, I/O buffers, disk caches, etc.
- **An attack could read any memory on the host, regardless of context.**

Impacted operating systems

All operating systems executing on a vulnerable CPU are subject to the vulnerability unless mitigation actions are taken.

Unimpacted Platforms

Not all compute platforms are impacted. Some platforms may be vulnerable in the future, but are not currently identified as having a viable proof-of-concept executed.

- Low-power embedded processors such as ARMv7, MIPS and similar. These CPUs don't use the feature used to drive the exploit.
- Legacy UNIX platforms such as PA-RISC, MIPS, Alpha.
  - These are likely vulnerable, but nobody has proven it yet.
- Some SPARC
  - Some SPARC not vulnerable by design
  - Oracle Statement released 18-Jan-2018 has very little detail but mentions patches for Solaris 10 and 11.
- Some IBM POWER[7]

[7] http://ibmsystemsmag.com/blogs/aixchange/january-2018/security-vulnerability-impacts-power-processors/

Timescale

While this issue hit public disclosure in December, some vendors have had considerable notice.

- At least six months for major vendors.
- Quite likely to be substantially longer in the security community.
- `http://bit.ly/2D42BW6`[8]
- There was even some mention of the potential in an ARM CPU programming manual from 2015 . . .

[8] https://www.theverge.com/2018/1/11/16878670/meltdown-spectre-disclosure-embargo-google-microsoft-linux

Mitigation Overview

- Mitigation will require operating system patches at a minimum.
- Further mitigation efforts will use CPU microcode updates.

OS patching

- Windows and Linux patching already released, version-dependant.
- Linux mitigation using two strategies:
  - `kpti` from mainline kernel team
  - `Retpoline` from Google.
- Linux with 3-series kernels not patched yet.
- Smaller platforms (FreeBSD, OpenBSD) did not receive advance notice and will take longer.
- Appliances (eg, NetApp) will take longer.
- Solaris x86 on 11 SRU28[9]
- **Particularly down-rev or out-of-support platforms are likely to never see a patch due to the kernel work required.**

[9] Pending 10 updates

Patching woes

- Despite advance notice, initial patch runs have met mixed success.
- First-round Windows patching rendered some AMD-based systems unbootable.
- Intel microcode patching has had similar problems with Linux.

Microcode, again

- Intel microcode updates have been causing widespread crashes as at 19-Jan-2018.
- Red Hat have pulled microcode updates from their patch releases.
- At this stage, all microcode updates should be tested on identical hardware in an SVT environment for quite some time before deployment.
- Some vendors are releasing BIOS patches with updated microcode. As these are often very hard to back out, they should be approached with extreme caution.

Performance impact

- Because the root cause of all these attacks is based in CPU performance features, software mitigations all carry a performance impact.
- At this point in time, performance impact is very hard to predict and based on CPU type and workload.
- I/O and network heavy loads are particularly vulnerable.
- Current-generation Intel CPUs will suffer less impact than previous generations.
- Patches should be tested extensively in model/test environments before widespread roll-out, preferably with similar hardware.
- Red Hat statement: https://access.redhat.com/articles/3311301

Recommended Actions

- Audit all hosts to determine vulnerability footprint.
- Hosts with low performance requirements or low workload should be patched when operating system patches are available.
- Performance critical workloads should be tested before applying fixes to production.
- The workarounds can be manually disabled, allowing normal patching cycles while avoiding performance penalties.

Workarounds

- If it is determined that the performance penalty is too severe, other strategies include:
  - Segregating performance-critical workload to firewalled or isolated hosts with stricter change review.
  - Deployment of host-IDS to track binary changes on hosts to ensure only trusted workloads running.
  - Re-locating workload to platforms that are not impacted.

Keeping ahead of the problem

- It is likely that most operating systems will see a long string of patches and mitigation efforts over the next year.[10]
- Hardware-fixed CPUs from Intel are most likely two years out.
- Vulnerability and patch revision management is going to be critical this year.
- Recommend using automated security tools to continually audit and report.
- In addition, security tools should be able to report on microcode updates.

[10] potentially longer...