OVER ONE MILLION HACKERS CAN NOW GET INSIDE YOUR FILES AND EMAILS WITH JUST TWO CLICKS OF THEIR MOUSE!

Microsoft, Google: We've found a fourth data-leaking Meltdown-Spectre CPU hole

Design blunder exists in Intel, AMD, Arm, Power processors

By Chris Williams, Editor in Chief 72 Reg comments SHARE ▼

A blowtorch burning a computer board

A fourth variant of the data-leaking Meltdown-Spectre security flaws in modern processors has been found by Microsoft and Google researchers.

These speculative-execution design blunders can be potentially exploited by malicious software running on a vulnerable device or computer, or a miscreant logged into the system, to slowly extract secrets, such as passwords, from protected kernel or application memory, depending on the circumstances.

Variants 1 and 2 are known as Spectre (CVE-2017-5753, CVE-2017-5715), and variant 3 is Meltdown (CVE-2017-5754). Today, variant 4

(CVE-2018-3639) was disclosed by Microsoft and Google researchers.

It affects modern out-of-order execution processor cores from Intel, AMD, and Arm, as well as IBM's Power 8, Power 9, and System z CPUs. Bear in mind, Arm cores are used the world over in smartphones, tablets, and embedded electronics.

ltdown graphic

The fourth variant can be potentially exploited by script files running within a program – such as JavaScript on a webpage in a browser tab – to lift sensitive information out of other parts of the application – such as personal details from another tab.

According to Intel, mitigations already released to the public for variant 1, which is the hardest vulnerability to tackle, should make attacks leveraging variant 4 much more difficult. In other words, web browsers, and similar programs with just-in-time execution of scripts and other languages, patched to thwart variant 1 attacks should also derail variant 4 exploits.

So far, no known exploit code is circulating in the wild targeting the fourth variant.

Another bug, CVE-2018-3640, was also disclosed: this is a rogue system register read, allowing normal programs to peek at hardware

nel-memory-
king Intel
cessor design
forces Linux,
dows redesign
READ MORE

status flags and the like in registers that should only really be accessible by the operating system kernel, drivers, and hypervisors.

How the fourth variant works

Variant 4 is referred to as a speculative store bypass. It is yet another "wait, why didn't I think of that?" design oversight in modern out-of-order-execution engineering. And it was found by Google Project Zero's Jann Horn, who helped uncover the earlier Spectre and Meltdown bugs, and Ken Johnson of Microsoft.

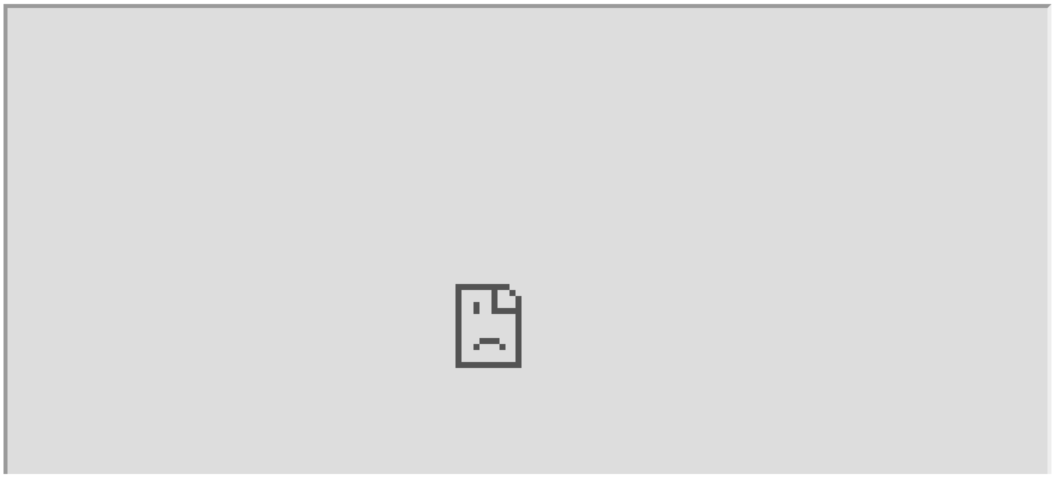
It hinges on the fact that when faced with a bunch of software instructions that store data to memory, the CPU will look far ahead to see if it can execute any other instructions out of order while the stores complete. Writing to memory is generally slow compared to other instructions. A modern fast CPU won't want to be held up by store operations, so it looks ahead to find other things to do in the meantime.

If the processor core, while looking ahead in a program, finds an instruction that loads data from memory, it will predict whether or not this load operation is affected by any of the preceding stores. For example, if a store is writing to memory that a later load fetches back from memory, you'll want the store to complete first. If a load is predicted to be safe to run ahead of the pending stores, the processor executes it speculatively while other parts of the chip are busy with other code.

That speculative act involves pulling data from memory into the level-one data cache. If it turns out the program should not have run the load before a store, it will unwind the instruction flow and restart the whole sequence – but it's too late. Part of the cache was touched based on the contents of the fetched data, leaving enough evidence for a malicious program to figure out that fetched data. Repeat this over and over, and gradually you can copy data from other parts of the application. It allows, say, JavaScript running in one browser tab to potentially snoop on webpages in other tabs, for instance.

The name Spectre was chosen deliberately: it is like observing a ghost in the machine. Private data can be discerned by watching the cache being updated by the processor's speculative execution engine. This speculation is crucial to running chips as fast as possible, by leaving as few processing units as idle as possible, but the downside is that the CPU can be tricked into revealing the contents of memory to applications and scripts that should be off limits.

A video lightly outlining the flaw, produced by Linux distro giant Red Hat, can be found below...



Intel, Arm, et al response

"Variant 4 uses speculative execution, a feature common to most modern processor architectures, to potentially expose certain kinds of data through a side channel," said Leslie Culbertson, Intel's executive veep of product security.

"In this case, the researchers demonstrated Variant 4 in a language-based runtime environment. While we are not aware of a successful browser exploit, the most common use of runtimes, like JavaScript, is in web browsers.

"Starting in January, most leading browser providers deployed mitigations for Variant 1 in their managed runtimes – mitigations that substantially increase the difficulty of exploiting side channels in a web browser. These mitigations are also applicable to Variant 4 and available for consumers to use today."

According to Culbertson, Intel and others will issue new microcode and software tweaks to more fully counter malware exploiting the fourth variant. These patches are being tested right now by computer and device manufacturers, we're told. Interestingly, they will be disabled by default when distributed to folks, presumably because the risk of a successful attack is so low. It's a tricky hole to fix, but also rather tricky to exploit. Another reason for the off-by-default state could be that Intel has struggled to put out stable Spectre updates in the past.

eption_screen
_648

"To ensure we offer the option for full mitigation and to prevent this method from being used in other ways, we and our industry partners are offering an additional mitigation for Variant 4, which is a combination of microcode and software updates," the exec said.

"We’ve already delivered the microcode update for Variant 4 in beta form to OEM system manufacturers and system software vendors, and we expect it will be released into production BIOS and software updates over the coming weeks.

"This mitigation will be set to off-by-default, providing customers the choice of whether to enable it or not. We expect most industry software partners will likewise use the default-off option. In this configuration, we have observed no performance impact. If enabled, we’ve observed a performance impact of approximately 2-8 per cent based on overall scores for benchmarks like SYSmark 2014 SE and SPEC integer rate on client and server test systems."

In a statement, a spokesperson for Arm told us:

need to go
per: Meltdown
Spectre flaws
force security
her down the
ck
READ MORE

> **This latest Spectre variant impacts a small number of Arm Cortex-A cores and is mitigated with an Arm-developed firmware update, which can be found at www.arm.com/security-update. As with previous published Spectre variants, this latest can only be executed if a specific type of malware is running on a user's device. Arm strongly recommends that individual users follow good security practices that protect against malware and ensure their software is up-to-date.**

We're also told that by July this year, Arm will make available to system-on-chip designers updated blueprints for Cortex-A72, Cortex-A73, and Cortex-A75 cores that are resistant to Spectre variant 2, and the Cortex-A75 will be updated to resist Meltdown, aka variant 3.

A spokesperson for AMD told us operating-system-level fixes – basically, some control registers need tweaking – should counter variant 4 attacks:

> **AMD recommended mitigations for SSB [the speculative store bypass] are being provided by operating system updates back to the Family 15 processors ("Bulldozer" products). For technical details, please see the AMD whitepaper. Microsoft is completing final testing and validation of AMD-specific updates for Windows client and server operating systems, which are expected to be released through their standard update process. Similarly, Linux**

> **distributors are developing operating system updates for SSB. AMD recommends checking with your OS provider for specific guidance on schedules.**
>
> **Based on the difficulty to exploit the vulnerability, AMD and our ecosystem partners currently recommend using the default setting that maintains support for memory disambiguation.**

Red Hat today published a substantial guide to the fourth variant, its impact, and how it works. VMware also has an advisory and updates, here, and the Xen Project explains itself and offers a fix, here. A spokesperson for IBM could not be reached for comment.

Context switch

We note that, so far, no malware has been seen attacking any of the Spectre and Meltdown holes in today's chips, let alone this latest variant, either because mitigations are widely installed making it largely fruitless, or it isn't worth the effort seeing as there are plenty of privilege-escalation bugs to exploit to get into a kernel and other applications.

This is despite various techniques emerging to exploit the Spectre family of design flaws, such as the ones revealed earlier this month, and twicein March.

Also, to exploit these flaws, malware has to be running on a device, which isn't always an easy task, unless you can trick a user into installing some bad code. Intel has proposed using graphics processors to scan physical memory for software nasties, such as Spectre-exploiting malware, during idle moments.

For us, these chip-level security bugs are a fascinating insight into the world of semiconductor engineering, where an intense focus on speed left memory protection mechanisms behind in the dust. And into the world of operating system and compiler design, where programmers are scrambling to secure kernels and user-mode code for years to come