Busting spectral kernel bugs!

Isolating the world's largest collaborative software project from vulnerabilities is no small task, as a Spectre-haunted **Mayank Sharma** discovers.

You won't have missed the news about Meltdown and Spectre – two of the most widespread security vulnerabilities that affects millions of CPUs in use today. They were discovered independently by teams at including Google Project Zero and researchers at the Technical University of Graz in Austria, among others.

Two things emerged once the dust settled on the implications, extent and severity of the vulnerabilities. One was the loud, unified criticism of how the disclosure about the vulnerabilities was communicated to the software stakeholders. The other was the stellar response by the Linux kernel community to mitigate the damage and contain the threats using software workarounds that compensate for the hardware vulnerability.

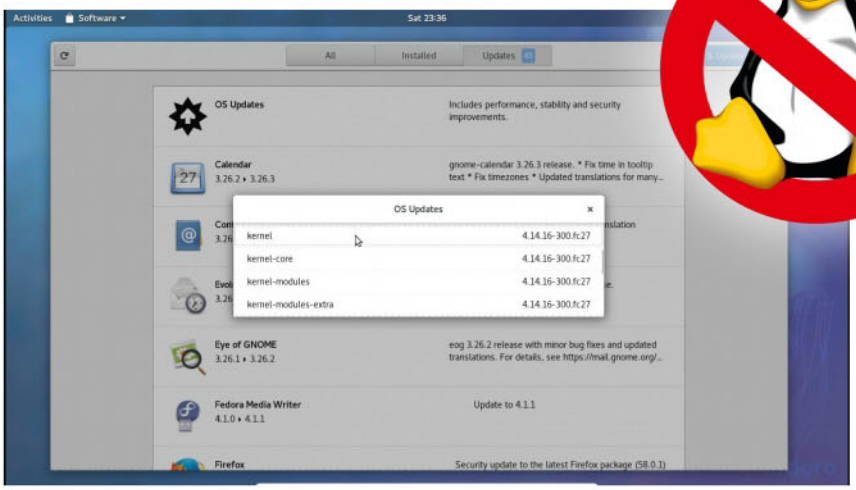
With more than 25 million lines of code spread over 61,000 files contributed by over 4,300 developers, the Linux kernel is the world's largest collaborative software project. The agility shown by the behemoth software in fixing a hardware flaw is commendable and deserves a closer look. Much like the kernel development itself, the process of insulating the kernel against vulnerabilities involves dozens of people spread all over the world. A core team of kernel developers form the Linux kernel security team that helps build a software moat around the kernel. They coordinate their efforts with those of several kernel security teams at the various marquee distribution projects that pitch in to test the patches before a vulnerability is made public. The process from the discovery of a vulnerability to patching the kernel with a fix happens at pace.

> **"Insulating the kernel against vulnerabilities involves people spread all over the world"**

While the Meltdown and Spectre vulnerabilities and their handling by everyone involved will be dissected at length, it gives us the perfect opportunity to take a peek behind the process and understand the efforts involved before the event notifier spits out the notice about a new kernel update.

Current Linux maintainer Greg Kroah-Hartman recently blogged about how the kernel team handles security threats. Greg notes that the Linux kernel community almost never declares specific changes as "security fixes". He explains that this is because of the difficulty in determining if a bugfix is a security fix or not, at the time of creation. Many bug fixes are only determined to be security related after much time has passed.

When security problems are reported to the kernel community, they're fixed as soon as possible (usually in about a week) and pushed out publicly to the development tree and the stable releases. Greg reasons that this is done to enable affected parties to update their systems before the reporter of the problem announces it.



Make sure you always install kernel updates whenever your distribution offers one.

Lifecycle of a vulnerability

Software projects usually track the list of common vulnerabilities and exposures (CVEs) to identify and fix vulnerabilities in software (*see the CVE box, below*).

"Every working day an Ubuntu security team member triages newly assigned CVEs. We scan MITRE's database, the oss-security mailing list, CVE lists maintained by other distros – including Debian, and the source repositories for many open source projects to identify newly assigned public CVEs," says Emily Ratliff, Head of Security at Canonical. The Security Team examines each CVE to determine whether Ubuntu is affected: "Each CVE is checked into our CVE database. Every working day, several Ubuntu Security Team members check the open issues in the database and prepare updates for release."

Eben Upton blogged in detail (http://bit.ly/pi-not-vulnerable) about the Raspberry Pi's invulnerability to Meltdown and Spectre.

CVEs aren't the usual starting point for fixing security issues at Fedora. In fact, Justin Forbes, one of Fedora's kernel maintainers, says that the two aren't even tied together: "Many CVEs are requested long after a fix has gone into mainline, or even many distro kernels." Justin reveals that the majority of them are not huge in themselves and of little consequence on their own, but "they still need to be fixed, as people can chain exploits to get much more with many small attacks."

Patch em' up

According to Justin, a potential security bug is usually found during a code review, or fuzz testing (a quality-assurance technique): "That bug is either fixed with a patch and then someone asks for a CVE, or the finder reports the bug (hopefully to the correct people) and someone ends up requesting a CVE and someone writes a patch to fix it." Justin adds that another frequent case is when a patch is written to fix a bug, and someone notices that the bug was actually a possible exploitable security issue. Discussions around fixing these issues happens in the public, he shares, usually across a combination of relevant upstream lists and security-focused lists.

Besides the vulnerabilities that comes up during code reviews, Justin points out that the kernel and the distributions also have to deal with issues for which a researcher has probably written proof-of-concept code and shown that they are easily exploitable, or can create large problems if exploited. He adds that these issues are frequently sent to various distribution security teams, or to specifically non-public email lists such as **security@kernel.org**.

Emily explains the process in more detail. She says that security researchers often disclose vulnerabilities to the Ubuntu Security Team privately via GPG encrypted email. When the vulnerabilities apply to open source projects, an Ubuntu Security Team member will coordinate with the researcher and the upstream communities to report the vulnerability to the project's developers.

"For vulnerabilities in projects originated or maintained by Canonical, the Ubuntu Security Team will file a private security bug in Launchpad and work with internal developers to get the vulnerability fixed," she says, adding that, "Canonical is the CNA (CVE Numbering Authority) for projects initiated by Canonical developers, so in this case, Canonical will assign a CVE to the vulnerability and notify MITRE of the details of the vulnerability."

Many issues are discussed privately before being made public. Emily points out that the »

What is a CVE?

CVE stands for Common Vulnerabilities and Exposures. It was brought to life in 1999 by MITRE, a non-profit organisation that operates research and development centres sponsored by the US federal government. The purpose of CVE is to identify and catalogue vulnerabilities in software or firmware inside a database in a bid for companies to improve their security. In other words, the aim of the CVE database is to standardise the way each known vulnerability or exposure is identified.

Each entry in the CVE database contains the standard identifier number along with a status indicator, a brief description and references to related vulnerability reports and advisories. The CVE database lists only publicly known vulnerabilities and exposures.

Organisations that identify and hand out CVE IDs for inclusion in first-time public announcements of new vulnerabilities are known as CVE Numbering Authorities or CNAs for short. Canonical, Red Hat and Mozilla are part of the 60-odd companies that have been designated as CNAs.

details of these issues are embargoed until an agreed upon Coordinated Release Date. There are many different ways that distributions and other affected parties come together to discuss these private issues. According to Emily large projects maintain their own lists of security teams who will be affected by security issues in the project. "One such list is the **security@kernel.org** mailing list, which discusses security issues in the Linux kernel," she says. "OpenWall maintains the distros list (**http://oss-security.openwall.org/wiki/mailing-lists/distros**), which is frequently used to discuss embargoed issues in a wide variety of open source packages."

Justin points out that when an issue is posted to **security@kernel.org**, coordination happens to get the patch developed and tested before such an issue is publicly disclosed: "The goal is to have fixes to users either before or immediately after disclosure to limit the exposure that users have. And of course, as a Linux distribution, the real goal is to have upstream fixed."

Kernel plumbing

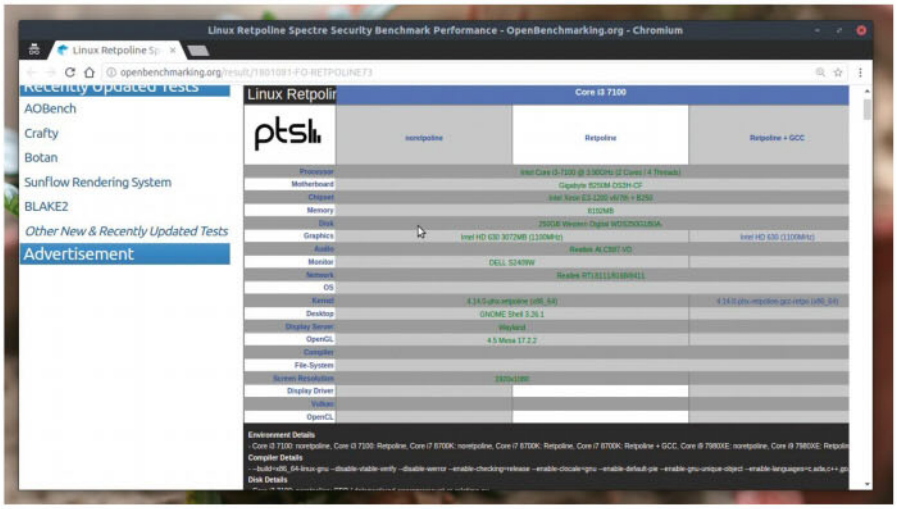
Once a vulnerability is found and reported, then the fix is generated in a similar manner to any other bug fix. Emily says that sometimes the reporter may include a patch or a test case reproducing the issue. The maintainers may accept the patch or produce their own fix. In some cases, she adds the distros produce patches and contribute the fixes back to the upstream projects.

Talking from the point of view of a distribution kernel, Justin says that their goal is to ensure that the users have as little security exposure as possible: "Because Fedora is so fast moving, we end up closing a lot of CVEs with 'we fixed that two weeks ago with kernel version x.y.z.'" He suggests that this is because a lot of these issues move through the upstream kernel before anyone requests a CVE or ties them in as security issues.

"For other issues, we'll often find a patch floating around which just hasn't made it into an upstream release yet," he adds. These patches are then pulled in and pushed out to users in the next kernel build. Justin reveals that the Fedora project pushes kernel updates almost every week. More serious issues are handled with a special build specifically to get the fixes out. However, he says that a lot of the CVEs are only questionably a security risk, and closer to regular bugs: "Things like a user with physical access to a machine can cause a DoS, or someone using a very uncommon piece of hardware can crash the system, etc. Those can wait for the regular build."

In case a patch for a vulnerability doesn't exist, Justin says that the project then follows up with developers who maintain that particular section of code, or write a fix and send it to them. He reveals that Fedora is very big on 'upstream first', and goes on to say that, "Ideally we would not be carrying any patches at all, because everything we need is upstream, so you'll never see a security patch sitting in a Fedora tree that hasn't been floated upstream. In the event of embargoed issues, we might test fixes internally, but they hit the Fedora tree at the time of public disclosure."

> **"The goal is to have fixes to users either before or immediately after disclosure"**



Phoronix.com has been benchmarking various components across distributions with the KPTI and Retpoline mitigation patches.

Marcus Meissner who is part of OpenSUSE's Security Team, sums up the entire process. "For embargoed issues, most of them happen via the distros and linux-distros' closed vendor coordination lists, where usually patches get posted as heads up and an embargoed date is agreed upon, with only some technical discussion. We also receive embargoed reports directly from some projects, like XEN, CURL and some others, usually accompanied with patches."

The twilight zone

Perhaps the most widespread security issues in recent times are CVE-2017-5754, CVE-2017-5753 and CVE-2017-5715 – dubbed Meltdown and Spectre. By most estimates the issues affect all computers built in the past decade irrespective of the operating system they run. The three different threats are not exactly the same, but they're related and use a similar exploit mechanism to gain access to privileged data. In a snap, the vulnerabilities involve reading memory locations that are supposed to be protected and reserved for use by the kernel. They exploit an architectural technique known as speculative execution, which was designed to improve computer performance.

Once the vulnerabilities were uncovered, all the stakeholders including the hardware vendors and the operating systems including several Linux distributions agreed upon 9 January, 2018 as the coordinated release date to disclose the vulnerability. On that date, they would all release updates to mitigate the issues. Due to several circumstances however, the issues were disclosed to the public ahead of schedule on 3 January, 2018. As a result, patches weren't available for some distributions when the vulnerabilities were disclosed.

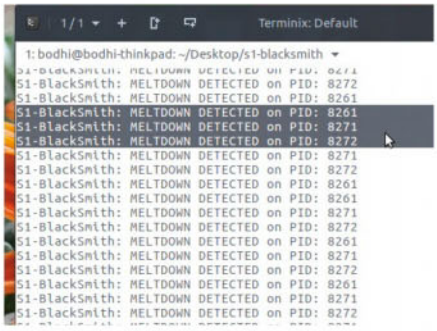
Work on addressing the vulnerabilities started appearing in the Linux kernel at the end of October with the KAISER set of patches. The KAISER patchset separates the page tables, which are currently shared between user and kernel space, into two sets of tables – one for each side. Subsequently, this work was renamed as kernel page-table isolation or KPTI. The patches were a fundamental change of the kernel's memory management function. Typically, such a major change would have been actively debated and

discussed. But since it was fast-tracked through the kernel releases, many people suspected that the work was being done under an embargo.

In a couple of days after the disclosure, all the fixes for Meltdown on the x86 hardware had been backported to the latest stable v4.14 kernel as well as in the v4.4 and v4.9 LTS kernel trees. Linus Torvalds released the first new Linux kernel of 2018, v4.15, on 28 January, after the longest development cycle for a new Linux kernel in seven years with nine release candidates.

Justin mentions that the Fedora developers knew that Meltdown was the most serious risk as it was fairly simple to exploit: "We were working with the KPTI patches over holidays to ensure that we were set to push fixes as soon as disclosure happened, and I think we got those out within a few hours of disclosure." He goes on to admit that there is still some follow up work with other architectures, but assures us that those are being fixed as quickly as possible, and have less exposure.

It was pretty much the same story over at Canonical. According to a blog post, Canonical was made aware of the vulnerabilities under embargo in November 2017 and their engineers were working "through the Christmas and New Years holidays, testing and integrating an incredibly complex patch set into a broad set of Ubuntu kernels and CPU architectures."



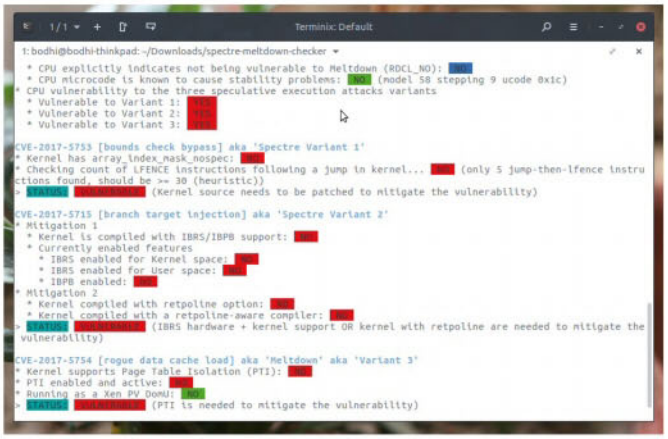
This freeware tool (http://bit.ly/sentinel-one) will alert you when attackers attempt to exploit the Meltdown vulnerability.

Of the two, Spectre is the trickier one and as Greg writes in a blog post was the last to be addressed by the kernel developers: "All of us were working on the Meltdown issue, and we had no real information on exactly what the Spectre problem was at all, and what patches were floating around were in even worse shape than what have been publicly posted." The Spectre issue is addressed in kernel v4.15 with the Retpoline code that was originally developed by Google.

Justin agrees that although Spectre is much more difficult to exploit it is still critically important to fix: "We followed upstream on this [Spectre], sometimes doing several builds a day to test new revisions of those patches. Again, x86_64 was the top priority because that is where the most users are." Laura Abbott, another Fedora kernel security engineer adds that since Fedora stays so close to upstream they were able to mostly take the patches as posted and just apply them to the Fedora kernel: "Distributions which were on older kernels had to do much more work to backport to their kernels."

Lines of communication

In terms of communication, Marcus explains that for deeper and longer running technical issues like with Meltdown and Spectre, ad-hoc communication channels / lists are established. For example, for Meltdown a special external mailing list had been set up, where the involved kernel developers were subscribed. Here they discussed integrating the mitigations and reviewed backports.



You can check whether your installation is vulnerable to Meltdown or Spectre with this script (http://bit.ly/speed-47).

There has been a fair amount of quibbling about how this process has played out (*see boxout, below*) and Linus Torvalds has been very vocal – as usual – in his criticism of Intel and its approach to the entire episode. Justin isn't impressed as well: "Clearly, the whole thing has been a bit of a mess, and I hope that we have learned something so that the next time something like this comes along, we can do a better job of coordinated disclosure."

While the immediate threat has been handled, Justin believes the vulnerabilities will haunt us for quite some time. "I think, due to the nature of these issues being architectural design and not simple code issues, this is something that will be ongoing for a bit."

"Plug the holes as quickly as possible, then optimise the fixes", is the way to move forward says Justin, commenting on the strategy of the kernel developers. Once the dust has settled and everyone's done pointing fingers, it is the community of open source developers that has once again saved the day. Justin sums it up perfectly: "No matter what distribution you are working with, or product you are manufacturing, the end result is that users need to be protected." LXF

Get off my lawn

There's been a lot of grumbling on the discovery to the communication and fixing. Greg shared his concerns in a blog post: "As for how this was all handled by the companies involved, well this could be described as a textbook example of how *NOT* to interact with the Linux kernel community properly. The people and companies involved know what happened, and I'm sure it will all come out eventually, but right now we need to focus on fixing the issues involved, and not pointing blame, no matter how much we want to."

Linus Torvalds was annoyed at Intel's attempt to pitch the vulnerability as a feature with an optional flag to disable it. "All of this is pure garbage. Is Intel really planning on making this s**t architectural?" Yet not everyone's happy with the Linux kernel's handling of the situation, either. Grsecurity's Brad Spengler said, "I think it points to either a massive lack of investment in security on behalf of the billion dollar companies involved in Linux, failure to retain the necessary talent, or a combination of both, that we seem to be the only ones doing this work, let alone without the benefit of months of advance notice in private."

And spare a thought for the *BSD camp, that was left scrambling for a solution. "We have received no non-public information. I've seen posts elsewhere by other *BSD people implying that they receive little or no prior warning, so I have no reason to believe this was specific to OpenBSD and/or our philosophy. Personally, I do find it....amusing?", wrote OpenBSD's Philip Guenther.