MUUGLines

The Manitoba UNIX User Group Newsletter

Volume 30 No. 6, February 2018 Editor: Trevor Cordes

Next Meeting: February 13th, 2018

iSCSI + S3 Cloud Backup

Wyatt Zacharias will present the AWS Storage Gateway with iSCSI block storage. The AWS Storage Gateway allows AWS S3 storage to be presented to hosts as native block devices using iSCSI. Wyatt will demonstrate the setup of a new storage gateway and how to connect a new volume to the gateway with iSCSI, and also talk about the costs and technical limitations of the service.

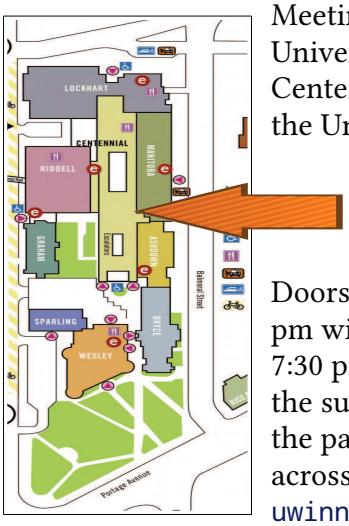
RTFM: jmtpfs & simple-mtpfs Android USB

Trevor Cordes will demonstrate how to transfer files to and from your Android device via USB cable using the MTP protocol and simple command line utilities. Cloud? We don't need no stinkin' cloud!

The latest meeting details are always at:
`https://muug.ca/meetings/`

Where to Find the Meeting

University of Winnipeg, Room 1C16A



Meetings are held in the University of Winnipeg's Centennial Hall, in the middle of the University Complex.

We can be found in room 1C16A.

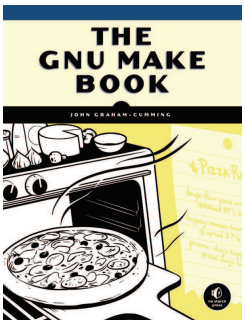
Doors are usually open by 7:00 pm with the meeting starting at 7:30 pm. Parking is available on the surrounding streets and in the parkade above the bus depot across Balmoral Street. See `uwinnipeg.ca/maps` for further information about parking and access to the campus.

Step Right Up, Get Yer Door Prizes!

Along with our usual e-book give away this month we will be giving away a physical book in all its dead-tree glory:

The GNU Make Book

Publisher: No Starch Press Release Date: 2015 256 pages



"The GNU Make Book demystifies GNU make and shows you how to use its best features. You'll find a fast, thorough rundown of the basics of variables, rules, targets, and makefiles. Learn how to fix wastefully long build times and other common problems, and gain insight into more advanced capabilities, such as complex pattern rules. With this utterly pragmatic manual and cookbook, you'll make rapid progress toward becoming a more effective user."

Creative Commons License

Except where otherwise noted, all textual content in this newsletter is licensed under a Creative Commons "Attribution-ShareAlike 2.5 Canada" License.

`http://creativecommons.org/licenses/by-sa/2.5/ca/deed.en_CA`

Meltdown/Spectre Update

As everyone in the industry now knows, the Meltdown / Spectre CPU-level vulnerabilities continue to be dealt with... and continue to cause problems. Yes, the sky doesn't seem to be falling (yet); but no, the bugs aren't a non-event either.

As of February 1st the initial live malware attacks are being seen in the wild. All seem to be based on the proof-of-concept examples provided by the security

researchers in January. All seem to be "feelers" to see what can be done with the flaws rather than actual, nasty, virulent attacks. But that could quickly change.

The worst part is that attacks are using JavaScript as the attack vector. That means that just browsing to an infected web site, or even viewing a trusted web site that contains an infected ad, can unleash the attack on your system.

The most likely targets of the attacks are your passwords / credentials for other websites, as well as other data that may be in browser memory, such as a credit card number you just used to buy something online. Think of the possible mayhem if attackers get your Paypal or online banking password.

As a fun bit of trivia, the majority of Intel/AMD CPUs (plus ARM and others, possibly to a lesser extent) produced **since 1995** are vulnerable. Basically anything since Pentium Pro. The feature causing the problem is "speculative execution", which is ubiquitous and responsible for massive performance gains in today's highly-pipelined CPUs.

The flaws are being addressed on many levels: Intel has released microcode and firmware updates at the hardware level. That said, they have recently paused and withdrawn some of these updates after reports of random reboots. OS vendors are integrating patches. Microsoft rolled out their initial patches but also had problems with random reboots and issued a hotfix that will disable the patches. Linux seems to be handling it the best, with measured and heavily-scrutinized patches that don't appear to be causing instability issues. The final level is browser vendors who are attempting to make it harder for timing attacks in general to be executed using JavaScript.

"OK, but I run the latest kernel version provided by my still-supported distro version, so I can ignore the whole thing." Not so fast: The higher 4.14 and new, **just released 4.15 Linux kernel** versions do contain initial patches to address both Meltdown and Spectre.

Meltdown ("Variant 3",CVE-2017-5754), the "easiest to fix" of the flaws appears to be mitigated with Kernel Page Table Isolation (KPTI) patches. (Meltdown affects only Intel CPUs.)

One catch, though, is that KPTI will slow you down. GKH says that 4.15 without KPTI is 7-9% faster than April 2017's 4.11, but 4.15 with KPTI is 1-2% slower than 4.11. That's a pleasant and slightly misleading way of saying KPTI'd 4.15 will be roughly 10% slower than the pre-KPTI 4.14 you were just running in December. And that's just for Meltdown, the easy bug!

Spectre ("Variant 2" CVE-2017-5715) is partially mitigated in Linux with the "Retpoline" patches. Think (and pronounce) like **trampoline**, using fudged stack **ret**urns in place of indirect target branches (the little gremlins that are causing the whole ruckus). Just as the Spectre 2 proof-of-concept code is fascinating, the retpoline counter-code assembler is equally so:

```
.macro NOSPEC_CALL target
jmp 1221f /*jumps to end of macro*/
1222:
push \target /*push ADDR to stack*/
jmp __x86.indirect_thunk
/* executes the indirect jump */
1221:
call 1222b /*psh ret addr to stack*/
.endm

/* the indirect_thunk */
    call retpoline_call_target
2:
    lfence /* stop speculation */
    jmp 2b
retpoline_call_target:
    lea 8(%rsp), %rsp
    ret
```

This is really slick and worth a few minutes to try to understand. So instead of the vulnerable jmp to ADDR (\target), you fake a subroutine call that sets (lea) the stack pointer to ADDR and then pretend **ret**urns to ADDR (but effectively jumps as the execution path was probably never near ADDR)! Spaghetti code at its finest.

The reason this apparent nonsense works is that the CPU speculative execution prediction always

assumes we are returning to 2:, and thus executes a lfence/pause (a pipeline stall/barrier instruction) repeatedly. This means that instead of a "maybe we'll jump, maybe not" affecting cache and allowing a side-channel timing attack, we have converted the code path into "we always jump or stop the pipeline", ensuring indirect target addresses are never prefetched into cache. This works because the return stack predictor is distinct from, and not vulnerable to gadget-injection due to cache poisoning like the branch predictor. Voila, you just solved a CPU problem with some tricky code.

Of course, this isn't free. You are effectively disabling speculative execution, and hindering one of the key features of modern pipelines, for this (common!) class of jump. So it will reduce execution speed.

Just to make your life more difficult, retpoline may not completely solve the problem, especially on Intel Skylake (i3/5/7/9 6xxx/78xx/79xx) and newer CPUs. They have a feature that if the Return Stack Buffer underflows the CPU falls back to Branch Target Buffer prediction, which retpoline does not protect. However, some have postulated that the balanced nature of the the stack manipulation may make said underflow rare.

Aside from retpoline, Intel has introduced a fix called IBRS in microcode (and submitted related fixes to the kernel). IBRS has been getting a bad rap. Linus has said this is "COMPLETE AND UTTER GARBAGE":

> *Intel is not serious about this, we'll have a ugly hack that will be so expensive that we don't want to enable it by default, because that would look bad in benchmarks*

And David Woodhouse says of IBRS:

> *So the part is I think is odd is the IBRS_ALL feature, where a future CPU will advertise "I am able to be not broken" and then you have to set the IBRS bit once at boot time to *ask* it not to be broken. That part is weird*

Lyft engineer Matt Klein reported a 20% slowdown due solely to IBRS being on. The jury is out on whether you need both retpoline and IBRS-in-microcode.

It is becoming clear there are legal, financial and marketing aspects to the whole Variant 2 saga, as well as technical. Some wildly speculate Intel may face lawsuits that could result in a recall of nearly every CPU still in use. Others point out how Intel is working with Linux (and presumably Microsoft) on 64-bit fixes, but leaving 32-bit land and BSDs in the lurch to fend for themselves.

As for Spectre Variant 1 CVE-2017-5753, it appears that recent gcc patches and a complete distro recompile (every userland binary as well as the kernel) may be enough… maybe. Since Variant 2's retpoline may require the same, it seems likely we'll see said distro-wide errata updates in the near future. (Pity Gentoo users.) This is a place where open-source software can shine: such a complete recompile is not just possible but somewhat easy for distros and users, where third party static binaries on systems are non-existent, or can be counted on one hand. Imagine the Windows world! However, even recompiles may not be enough to protect from malicious externally-supplied intraprocess code in JIT compilers, which are used in more places than you'd guess.

All hysteria aside, even with all fixes in place, most home users will not experience enough of a performance degradation to really notice. Business users, HPC, cloud instance consumers, and gamers, on the other hand, may get a bit miffed. Once every mitigation is in place, they could see a noticeable drop-off in performance. However, since all the current fixes are in software or microcode, expect to see tweaks and optimizations added over time.

What about phones/tablets? This is perhaps the worst part, and very few are focusing on it. Android and iOS are basically built on a sandbox model where apps are supposed to be cordoned off but they all (mostly) run as a single non-root user, and so are ripe for the cross-process same-user aspects of Spectre – the same as browsers (sort of).

The difference is phone vendors are atrocious at updating the OS of even slightly older phones. Most phones that have a release date (not a bought-date!) of more than one to three years ago do not receive any OS updates. So while you may be using a modern, patched browser on your older phone, and gain some protection from js exploits, you cannot trust apps to not contain an exploit, even ones previ-

ously vetted. If exploits become common, and vendors don't release updates for anything but the latest generation, it's basically throwaway time.

Back to desktops, Intel and AMD have recently promised that their next-gen CPUs will contain fixes in hardware. Intel says "later this year", so that means possibly in Cannon/Ice Lake. AMD says Zen2 (so not Zen+). Hopefully the ideas used to fix these bugs in hardware are made public as they would make fascinating reading. Alleviating these problems without the concomitant performance hits makes for challenging CPU design. In the meantime, js-blockers for browsers, both desktop and mobile, are looking like must-haves now.

Consolidated source links:

`https://tecnopolis.ca/muug/meltdown.html`

O'Reilly Books: Latest News

In the continuing saga of O'Reilly Media, computer book publisher extraordinaire, there have been some interesting developments. First, their customer support appears to be in disarray. While their online community support page is still available and being used, it appears currently not to be linked to from anywhere on their main site, so only those in the know know: `http://support.oreilly.com/oreilly`

The "Bring back PDF ebooks" topic, mentioned in an earlier MUUGLines, still garners a (negative) reply every few weeks, and is up to 61 replies and 39 followers. Also, the level of engagement from ORM employees seems vastly diminished.

Second, if you manage to get to their old home page, by clicking *shop* from their new "Safari Yay!" home page, you'll find they have not updated their "New Books" scrollers since September 2017. This is even after their customer support has been notified, and has responded with a "we'll look into it" on the aforementioned support site. There is currently no way to see a listing of their new releases without delving into Safari, making it very difficult to eyeball and wish list future purchases.

Third, at the same time as their massive shop shutdown they ceased publishing their nine periodic (mostly weekly / bi-weekly) newsletters. There was no email notice; they just stopped. The subscribe page is still linked to from their site, and you can still subscribe, but nothing is ever sent. The samples on the page are all dated from the same time as well.

In sum, something massive appears to have occurred June-August 2017 resulting in this massive shakeup, and ORM is completely underplaying the entire episode. While their higher-ups have released a smattering of public missives on the subject, none give much insight into what happened and what their plan is going forward… other than "Safari Yay!". If one had to speculate, one would think they had to massively scale back resources for whatever reason, and indiscriminately slash features, offerings and support. (Interestingly, their conference calendar remains untouched: packed with over a dozen in 2018 alone.)

As for Amazon book purchasing updates: Prices on used ORM (and related) books remains extremely high. In many cases, the sold-by-Amazon offerings are the same price or cheaper. Under $10 deals remain rare, let alone the old 1 cent ones from yesteryear. If you can find an under-$20 including shipping, you're doing pretty well.

Amazon has abandoned the forced $6.49 S&H price, much to everyone's chagrin, because now the best-used-price shown on a book's main page is useless, as it is often $something-low with $19 S&H. Great, another eBay. Once you go to the sub-page it sorts by total price, but that's not useful for quick wish list scans.

Worse still, now the "new" price shown cannot be trusted because Amazon will list as the main, Amazon price prices from other sellers when their price undercuts Amazon. And that price usually does not qualify for free shipping, meaning the price is actually $6.49 (or more) higher. This makes quick scans of even the "new" price on your wish list much less useful. It also begs the question: how are these other vendors undercutting Amazon's price on **new books** in quantity?