Security Now! #715 - 05-21-19

CPU.fail

This week on Security Now!

As expected after last week's Tuesday morning end-of-embargo on details of the next round of Intel processor information leakage problems, we will take a closer look at the new challenges they create and the impact of their remediation on system performance and stability. But before that we look at last Tuesday's patches from Microsoft, Abode and Apple. We examine a new big security problem for Cisco that ever has stock analysts taking notice, we check-in on the ongoing troubles with the cryptocurrency market, see what Johns Hopkins associate professor Matthew Green tweeted about the trouble with Google's Titan Bluetooth dongle, and deal with yet another monthly problem with Windows 10 updates. We then touch on a bit of miscellany and wrap up with a look at the new so-called Microarchitectural Data Sampling vulnerabilities.

It's easy to dismiss this photo and the similar photos we've seen. But, really, stop to consider that this actually exists somewhere. Someone placed a motorized automobile blocking arm across the road to prevent anyone from passing. And it could not be more clear how well that has worked out for those wishing to control access...

Security News

Windows XP gets a patch!

You KNOW something is worrying Microsoft deeply when they are willing to reach back to Windows XP and produce a patch for that long-since-deserted operating system.

https://blogs.technet.microsoft.com/msrc/2019/05/14/prevent-a-worm-by-updating-remote-desktop-services-cve-2019-0708/

Simon Pope, Director of Incident Response with the Microsoft Security Response Center (MSRC) titled his announcement posting as plea, with the headline: "Prevent a worm by updating Remote Desktop Services"

Today Microsoft released fixes for a critical Remote Code Execution vulnerability, CVE-2019-0708, in Remote Desktop Services – formerly known as Terminal Services – that affects some older versions of Windows. The Remote Desktop Protocol (RDP) itself is not vulnerable. This vulnerability is pre-authentication and requires no user interaction. In other words, the vulnerability is 'wormable', meaning that any future malware that exploits this vulnerability could propagate from vulnerable computer to vulnerable computer in a similar way as the WannaCry malware spread across the globe in 2017. While we have observed no exploitation of this vulnerability, it is highly likely that malicious actors will write an exploit for this vulnerability and incorporate it into their malware.

[- I LOVE the fact that the world is finally sitting up and taking sober responsibility for these sorts of problems. We know that anyone can make a mistake, and that such mistakes can go unseen for years, and sometimes even decades. It's not the fact of the mistake that matters, it's how those who are now responsible for fixing it deal with that responsibility. So this is another "Bravo" for Microsoft. On the other hand... they didn't really have much choice. -]

[Simon continues....] Now that I have your attention, it is important that affected systems are patched as quickly as possible to prevent such a scenario from happening. In response, we are taking the unusual step of providing a security update for all customers to protect Windows platforms, including some out-of-support versions of Windows.

Vulnerable in-support systems include Windows 7, Windows Server 2008 R2, and Windows Server 2008. Downloads for in-support versions of Windows can be found in the Microsoft Security Update Guide. Customers who use an in-support version of Windows and have automatic updates enabled are automatically protected.?

Out-of-support systems include Windows 2003 and Windows XP. If you are on an out-of-support version, the best way to address this vulnerability is to upgrade to the latest version of Windows. Even so, we are making fixes available for these out-of-support versions of Windows in KB4500705.

Customers running Windows 8 and Windows 10 are not affected by this vulnerability, and it is no coincidence that later versions of Windows are unaffected. Microsoft invests heavily in strengthening the security of its products, often through major architectural improvements that are not possible to backport to earlier versions of Windows.

There is partial mitigation on affected systems that have Network Level Authentication (NLA) enabled. The affected systems are mitigated against 'wormable' malware or advanced malware threats that could exploit the vulnerability, as NLA requires authentication before the vulnerability can be triggered. However, affected systems are still vulnerable to Remote Code Execution (RCE) exploitation if the attacker has valid credentials that can be used to successfully authenticate.

It is for these reasons that we strongly advise that all affected systems – irrespective of whether NLA is enabled or not – should be updated as soon as possible.

I've said several times recently, when we've been covering other instances of Remote Desktop Protocol commandeering is Internet-exposed systems, in those cases presumably using brute force authentication guessing (now referred to as Credential Stuffing) that there is simply no safe way to have an RDP server publicly exposed. None.

RDP runs over a well-known port (3389) so scanning the Internet's IPv4 address space for them is trivial. About 11 million IP addresses accept connections on 3389 and of those, somewhere around 4.1 million respond as RDP servers. That's insane. And it certainly explains and warrants Microsoft's deep concern when we absolutely know that some probably-significant percentage of those will be WinXP, Win7, Server 2003, 2008 or 2008R2 machines.

Anyone who wishes to make Remote Desktop Protocol available remotely should, or really MUST, place it behind a VPN. Simply run an instance of an OpenVPN server using certificate-based authentication. Setup an instance of OpenVPN client in any authorized roaming client machines and give them individual certificates signed by the server's private key. You then have a robust means of accessing RDP remotely and also of controlling and monitoring who has access.

However, there are many millions of probably-exploitable RDP servers that will probably not be updated. If auto-update is enabled and if the machines are setup to reboot unattended, then perhaps this problem will be self-healing. That remains to be seen. But since the only way this "installed base" of millions of potentially vulnerable RDP servers have been hackable until now has been by guessing logon credentials, there would be HUGE pressure to reverse engineer the changes made to prevent this pre-authentication bypass and then create, if not a worm, then at least a great many more potentially powerful Bots for mischief.

Back on September 1st of 2017, the website "TechGenix" concluded a posting about this by writing: "Ultimately, it is a race against time to secure these exposed RDP machines, all 4.1 million of them, as any zero-day employed on a mass scale could allow for mass remote hijacking or malware deployment against powerful organizations."

Despite Microsoft's best efforts at emergency remediation, that day is, to some extent, now very likely to arrive before long.

An Important Patch Tuesday (actually it was kinda CRITICAL)
Microsoft and Adobe both released regularly scheduled patches last Tuesday.

This month's Microsoft Patch Tuesday addresses 79 vulnerabilities with 22 of them labeled as Critical. Of the 22 Critical vulns, 18 are for scripting engines and browsers. The remaining 4 are remote code execution (RCE) in Remote Desktop, DHCP Server, GDI+, and Word.

Microsoft also released guidance for the recently disclosed Microarchitectural Data Sampling (MDS) techniques -- as they are now being termed -- known as ZombieLoad, Fallout, and RIDL. We'll be discussing them in some detail at the end of this podcast. Adobe, for it's Patch Tuesday, fixed continuing problems in Flash, Acrobat/Reader and their Media Encoder. While the Flash patch fixed only one known flaw, and the Media Encoder patches fixed two, the Acrobat/Reader patches closed a whopping 83 vulnerabilities.

But as for Windows, the remote code execution vulnerabilities occurring in Windows Scripting Engine, Browser, GDI+, and Word patches make updating workstations a priority.

The fourth Remote Code Execution vulnerability, CVE-2019-0725, affects Windows' DHCP Server is ranked Critical. Any unauthenticated attacker who can send packets to a DHCP server can exploit this vulnerability. This patch should be prioritized for any Windows DHCP implementations. If DHCP problems ring a bell it's because both February and March also had DHCP vulnerabilities patched.

One of the 22 other vulnerabilities that isn't a remote code execution vulnerability can lead to local unauthorized code execution. It is an elevation-of-privilege vulnerability known to be actively exploited in the wild tied to the Windows Error Reporting feature and is being abused by attackers who have gained local access to unpatched PCs. It is of value to malware which is able to use it to obtain arbitrary code-execution in kernel mode — resulting, as we know, in complete system compromise.

For the Zero Day Initiative blog post last Tuesday, Dustin Childs, Trend Micro's communications manager wrote: "They would need to first gain access to run code on a target system, but malware often uses elevations like this one to go from 'user' to 'admin' code execution. While details about the use of the exploit are not available, it is likely being used in limited attacks against specific targets."

Cisco gets in trouble with "Thangrycat"
A new and very serious vulnerability present in hundreds of millions of Cisco routers, switches and firewalls has recently come to light.

InvestorPlace.com notes "Cisco Earnings Were Great — But Beware Thangrycat. Cisco's latest security weakness is something to consider if looking to buy." They go on to say that they have no idea what to make of "Red Balloon's" description of the massive sweeping vulnerability that's apparently present in hundreds of millions of Cisco's deployed routers, switches and firewalls (what's a router?).

But we, here, know exactly what it is that routers route. So… Security researchers with the firm "Red Balloon" have identified a vulnerability and have received the CVE of 2019-1649. Dubbed "Thangrycat" and, like the artist formerly known as Prince whose name is a symbol, Thangrycat isn't its official name… it's three very unhappy looking cat face icons.

Thangrycat affects multiple Cisco products that support Trust Anchor module (TAm). As its name suggests, the Trust Anchor Module is a critical component of Cisco's hardware-based Secure Boot functionality which has been implemented in nearly all of Cisco's enterprise devices since 2013. It promises to ensure that the firmware running on hardware platforms is authentic and unmodified. Unfortunately, it turns out that's a promise it's unable to keep.

https://thrangrycat.com/

In the words of its discoverers...

Red Balloon Security, Inc. is disclosing two vulnerabilities affecting the products of Cisco Systems, Inc. ("Cisco"). The first, known as 😾😾😾 allows an attacker to fully bypass Cisco's Trust Anchor module (TAm) via Field Programmable Gate Array (FPGA) bitstream manipulation. The second is a remote command injection vulnerability against Cisco IOS XE version 16 that allows remote code execution as root. By chaining the 😾😾😾 and remote command injection vulnerabilities, an attacker can remotely and persistently bypass Cisco's secure boot mechanism and lock out all future software updates to the TAm.

😾😾😾 is caused by a series of hardware design flaws within Cisco's Trust Anchor module. First commercially introduced in 2013, Cisco Trust Anchor module (TAm) is a proprietary hardware security module used in a wide range of Cisco products, including enterprise routers, switches and firewalls. TAm is the root of trust that underpins all other Cisco security and trustworthy computing mechanisms in these devices. 😾😾😾 allows an attacker to make persistent modification to the Trust Anchor module via FPGA bitstream modification, thereby defeating the secure boot process and invalidating Cisco's chain of trust at its root. While the flaws are based in hardware, 😾😾😾 can be exploited remotely without any need for physical access. Since the flaws reside within the hardware design, it is unlikely that any software security patch will fully resolve the fundamental security vulnerability.

We've not talked about FPGA bitstream programming in the past. FPGA stands for "Field Programmable Gate Array" and it's a means for creating "soft hardware". As its name suggests, an FPGA is a massive array of logical elements which can be dynamically configured --

programmed -- by loading a "bitstream" into the device. So the FPGA device itself is blank, unconfigured and generic when it's first powered up. But it's able to "suck in" a programming bitstream from a non-volatile EPROM to configure itself. In this instance it appears that Cisco made a serious mistake of not protecting some aspects of this process from tampering by a determined attacker.

From their Q&A answering the question: "How widespread is this?" they explain: This vulnerability affects Cisco products with an FPGA based TAm. Cisco released their list of more than 100 product families with this vulnerability. (And, I have to say, it's truly horrifying. Scrolling down this list is sobering in the extreme.)

Q: What are the implications of demonstrating modification of the FPGA bitstream?

A: Our findings support the practical exploitation of FPGA-based devices via direct bitstream analysis and modification. Through our research we developed a series of techniques to reliably add, subtract, and alter FPGA behavior without any need to perform register-transfer level (RTL) reconstruction. By demonstrating successful FPGA modification on the Xilinx Spartan 6 LX45T, we find that our bitstream manipulation techniques present a range of potential applications for persistent FPGA implants, physical destruction of embedded systems, and attacks against FPGA-based systems, such as software-defined radios, advanced automotive driver assist modules, weapon guidance systems, and more.

Q: Have these vulnerabilities been exploited in the wild?

A: We are unaware of any use of this exploit in the wild, but the potential danger is severe.

Q: What action can be taken?

A: Please consult Cisco’s official security advisory. We did not receive early access to Cisco’s security patch, and will be analyzing the patches as they are made publicly available. Since 😾😾😾 is fundamentally a hardware design flaw, we believe it will be very difficult, if not impossible to fully resolve this vulnerability via a software patch.

Their justification for naming this attack with three emoji's was so whacky that they should have simply said "Because we wanted to... Meow!" However, their Q&A asked themselves the question: "How do you pronounce this vulnerability name?" to which they explained:

A: There is no phonetic transcription for this specific sequence of repeated emojis, and the pronunciation is open to interpretation. We suggest “Thrangrycat” as a suitable enunciation."

These guys will be showing off their work at this summer's BlackHat 2019.

https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-20190513-secureboot

I was unimpressed with Cisco's own vulnerability disclosure because, doubtless in something of a panic, they downplayed the problem's severity by stating:

"A vulnerability in the logic that handles access control to one of the hardware components in Cisco's proprietary Secure Boot implementation could allow an authenticated, local attacker to write a modified firmware image to the component. This vulnerability affects multiple Cisco products that support hardware-based Secure Boot functionality."

Uh huh. The only problem is that these guys show the chaining of this flaw with another remote command injection vulnerability against Cisco's widely used IOS XE version 16 that allows remote code execution as root. In other words... Any remote attacker can leverage these two vulnerabilities in chain sequence. So, yes, technically the FPGA nightmare is local only, but it's possible for a remote attacker to obtain the local privileges needed.

Cisco continues: *"The vulnerability is due to an improper check on the area of code that manages on-premise updates to a Field Programmable Gate Array (FPGA) part of the Secure Boot hardware implementation. An attacker with elevated privileges and access to the underlying operating system that is running on the affected device could exploit this vulnerability by writing a modified firmware image to the FPGA. A successful exploit could either cause the device to become unusable (and require a hardware replacement) or allow tampering with the Secure Boot verification process, which under some circumstances may allow the attacker to install and boot a malicious software image."*

Again, that's not patently false on its face, but it doesn't faithfully convey the nature of the danger this flaw represents, and isn't that the whole point of a vulnerability disclosure?

And, again, Cisco's success as a vendor of networking gear is undeniable. In this case, as a result, we're talking about hundreds of millions of devices exposed on the public Internet.

Cryptocurrency Hacks Still Growing — What Does That Mean for the Industry?
https://thehackernews.com/2019/05/bitcoin-ethereum-hacks.html

We've been talking a lot about Coinhive and browser-based mining of Monero. We haven't check in on the larger CryptoCurrency world recently. The HackerNews wrote up an interesting summary snapshot with information from several sources including the CipherTrace 2018 cryptocurrency report. So, with a bit of paraphrasing for length, I want to share this nice write up by The Hacker News:

Though once synonymous with underground networks and black hat hackers, bitcoin and other cryptocurrencies have gone mainstream over the past two years. In 2017, we saw the skyrocket of bitcoin to an all-time high of close to $20,000 followed by a significant decline the following year. But beyond the ups and downs in the market for the world's largest cryptocurrency is a much more sinister story revolving around cyber-attacks of the economy's newest asset class.

In 2018, it estimated that as much as $1.7 billion worth of cryptocurrencies were swindled away from investors (likely more) through a variety of means. Whether accomplished through hacking, phishing, or other forms of scamming, it's clear that the crypto industry is facing a serious dilemma with security. For a technological movement based on decentralization and the advantages it offers for security, the number of breaches occurring is startling.

Cryptocurrencies offer users a way to send money without the need for a third party, yet the industry as a whole is dealing with more security vulnerabilities than centralized financial firms doing the same thing. During the same time period, more traditional companies that transfer money and banks have seen nowhere near the same amount of issues with hackers. So, what's the problem?

While cryptocurrencies and blockchain technology are decentralized in nature, there are many aspects of the cryptosphere that aren't. The number one culprit in 2018 was cryptocurrency exchanges. Unlike the underlying technology behind currencies like bitcoin, Ether, and Litecoin, cryptocurrency exchanges are centralized in nature and not yet regulated to the same extent as most financial firms.

According to data from CipherTrace's 2018 cryptocurrency report, $950 million of the total $1.7 billion stolen were from exchanges and infrastructure services. Exchange services are a particular pain point for the industry because they're one of the easiest ways for users to get started with cryptocurrencies as some even handle fiat currency.

Often referred to as "on-ramps" for the crypto industry, fiat friendly exchanges are easy for beginners to use and purchase their first crypto. However, with that ease of use comes a major target for hackers and phishers. 2018 was undoubtedly a big year for cryptocurrency hacks, setting new records for theft, but 2019 may not be far off. Just four months into the new year, here are two of the major cyber attacks that have already occurred in 2019.

Last month, popular South Korean exchange Bithumb announced that it suffered a security breach and theft to the tune of $19 million worth of cryptocurrencies, making it the largest of the year. The exchange suspects that the attack may have been carried out with the help of an insider to steal EOS and XRP. More notable is that this is not the first time the exchange has been compromised. In 2017, hackers managed to get away with $31 million worth of cryptocurrencies from the exchange and around $1 million the year before.

Also occurring in March, Singapore-based cryptocurrency exchange DragonEx revealed that it too had been hacked. After going public with the announcement, the company revealed that it estimates somewhere around $7 million worth of cryptocurrencies were stolen and transferred off the exchange to various other exchanges and wallets. DragonEx has stated that it's working on a preliminary compensation plan for clients whose funds were stolen and has denied rumors of potential bankruptcy. The exchange has publicly released the addresses of wallets it believes to be possibly holding stolen funds and has asked for assistance from other exchanges and wallet providers.

As the total amount of cryptocurrencies stolen from investors continues to rise each year, security experts are scrambling to find the most effective methods to combat hacking. However, also included in CipherTrace's report was the changing landscape of crypto hackers and scams. In the first three quarters of 2018, the majority of theft happened via direct exchange hacks, now that's starting to change.

With exchange services beginning to take security concerns more seriously, different forms of attacks are becoming more common. Looking forward into 2019, experts suggest that tactics like social engineering and the utilization of insiders may be the largest threats. Scammers and

phishers see the cryptocurrency space as low hanging fruit over recent years as more newcomers flock to the scene. Between fraudulent social media accounts claiming to be influential people to bogus "exchange support members" accounts claiming to help with logging in problems, the way the industry thinks about security breaches is changing.

Matthew Green -vs- Google

So, Matthew Daniel Green, our oft-quoted cryptographer and security technologist who is an Associate Professor of Computer Science at the Johns Hopkins Information Security Institute got himself a bit worked up over the details of the recent flaw discovered in the configuration of Google's TITAN Bluetooth security dongles.

How do we know? Well, Matthew tweeted...
https://twitter.com/matthew_d_green/status/1128706561164173314?ref_src=twsrc%5Etfw

Quoting from the Bluetooth specification:

The encryption key (K_C) is derived from the current link key and may vary in single byte increments from 1 byte to 16 bytes in length, as set during a negotiation process that occurs between the master and slave devices. During this negotiation, a master device makes a key size suggestion for the slave. The initial key size suggested by the master is programmed into the controller by the manufacturer and is not always 16 bytes. In product implementations, a "minimum acceptable" key size parameter can be set to prevent a malicious user from driving the key size down to the minimum of 1 byte, which would make the link less secure.

Matthew: *"Like, what kind of idiot protocol lets users negotiate a "maximum key size" that can be as small as 1 byte. (A default that, fortunately, should be higher in recent versions.)"*

He continued: "Don't rely on Bluetooth security protocols for anything, ever. Just treat them like a particularly inefficient version of Base64 encoding."

And: *"What would you possibly do with a 1-byte key? Is anyone at Bluetooth SIG even **awake** when these idiot decisions get made?"*

"Hey maybe we should call some experts and ask them if 1 byte is a good minimum size for an encryption key?" "Nah, we've got all the expertise we need right here. That's 256 whole keys."

Then he tweeted: *"Bluetooth is the Michael Bay movie of encryption protocols."* And I thought… Huh?...

Then he explains his reference: *"It doesn't make any f**king sense, and it explodes."*

And then someone tweeting as "Tibor Jager" who is enjoying Michael's tirade adds: *"I like the way they use the phrase "less secure"."*

To which Matthew tweets in reply: *"What could it possibly be less secure than?!"*

So what’s Matthew so worked-up about?

It seems that a team of security researchers at Microsoft discovered a serious vulnerability in the Bluetooth version of Google's Titan Security Keys that can not be patched with a software update. So Google has announced that it will replace all affect keys at no charge.

In their security advisory published Wednesday, Google said a "misconfiguration in the Titan Security Keys Bluetooth pairing protocols" could allow an attacker who is physically close to your Security Key (~within 30 feet) to communicate with it or the device to which your key is paired.

[And we know from Matthew that this “misconfiguration” allows the use of a 1-byte key.]

As we covered at the time last summer, the Titan Security Key system consists of a USB dongle to provide hardware-based two-factor authentication for online accounts which require the highest level of protection against phishing attacks. Selling for $50 in the Google Store, the system includes a pair of keys—a USB security key with NFC, and a battery-powered, Micro-USB-equipped Bluetooth/NFC key—for multi-factor authentication.

According to Google, the vulnerability only affects the BLE version of Titan Security Keys that have a "T1" or "T2" sign on the back. Other non-Bluetooth security keys, USB or NFC supported versions, are safe to use.

The attack scenarios Google Cloud Product Manager Christiaan Brand described in a blog post:

"When you're signing into an account on your device, you are normally asked to press the button on your BLE security key to activate it. An attacker in close physical proximity at that moment in time can potentially connect their own device to your affected security key before your own device connects. In this set of circumstances, the attacker could sign into your account using their own device if the attacker somehow already obtained your username and password and could time these events exactly."

"Before you can use your security key, it must be paired to your device. Once paired, an attacker in close physical proximity to you could use their device to masquerade as your affected security key and connect to your device at the moment you are asked to press the button on your key. After that, they could attempt to change their device to appear as a Bluetooth keyboard or mouse and potentially take actions on your device."

Microsoft originally discovered the vulnerability and disclosed it to Google, as well as Feitian, the company that makes Titan Keys for Google and also sells the same product (ePass) under its own brand.

Feitian also made a coordinated disclosure about this vulnerability the same day as Google and is offering a free replacement program for its users.

Google says that since the issue only affects the Bluetooth Low Energy pairing protocol and not the cryptographic security of the key itself, affected users should continue using their existing keys until they get a replacement.

Google also says that using the less-than-fully-secure Bluetooth security key is still more secure than turning it off altogether or relying on other two-factor authentication methods like SMS or phone call.

iOS v12.3
This update included fixes to 42 CVEs, 20 of which, not unusually, affected WebKit.

As we know, Apples tends to be the most tight-lipped about details of their vulnerabilities and the ones that jump out usually involve a vulnerability that might allow a remote attacker or local app to take control of the device at some level – like most of the WebKit flaws. But it's difficult to be sure.

There's CVE-2019-8585 which affects "CoreAudio." The suggestion is that it might give malware a route to compromise using a malicious movie file. That's serious if, as it seems, it wouldn't necessarily require the target victim to do anything… if, indeed, the attack was targeted and not sprayed.

Then there was CVE-2019-8593 in AppleFileConduit, and CVE-2019-8605 in the kernel, either of which might allow an app to gain system privileges, or CVE-2019-8637 in AppleFileConduit, through which -- Apple reluctantly admits -- a "malicious application may be able to gain root privileges." Those would require users to download malicious apps.

So, as usual with Apple, we don't know much more but they're fixing things and, in general, iOS is a comparatively safe place to be.

Meanwhile, a new problem surfaced with Windows 10 after last Tuesday's updates.
Microsoft has acknowledged that a known Windows 10 problem is causing some users' systems to freeze. The issue arises when users who created a restore point before updating subsequently attempt to back out of the updates -- which should be fine since, after all, that's the whole point of having system restore checkpoints to fall back to.

But when users attempted to do this they were met with a "Stop error" that blocked them from proceeding, booting, and getting their machines back. Microsoft's advisory said that "this is a known issue in Windows 10" and offers workarounds to the problem. But what an annoyance.

https://support.microsoft.com/en-us/help/4503117/system-restore-on-windows-10-fails-after-you-install-updates

Title: "You cannot restore the system to a restore point after you install a Windows 10 update"

Consider the following scenario:
- You install Windows 10 on a clean computer.
- You turn on system protection, and then create a system restore point that is named "R1."
- You install one or more Windows 10 updates.
- After the updates have finished installing, you restore the system to the "R1" restore point.

In this scenario, the system is not restored to the "R1" restore point. Instead, the computer experiences a Stop error (0xc000021a). You restart the computer, but the system cannot return to the Windows desktop.

Up at the top of the advisory page which explained how, after installing and deciding to back out of Windows 10 updates a Windows 10 system would be bricked, I noticed the helpful reminder that "After 10 years, support for Windows 7 is nearing the end." Aren't we glad that Microsoft has saved us from that crapware-free, non-monitored, much simpler operating system that just works and whose updates haven't had any monthly troubles such as have plagued Windows 10 nearly every month since last October?

Anyway... In this advisory under "Cause" they continue: “This is a known issue in Windows 10.”

[Oh, well then...]

During the system restore process, Windows temporarily stages the restoration of files that are in use. It then saves the information in the registry. When the computer restarts, it completes the staged operation.

In this situation, Windows restores the catalog files and stages the driver .sys files to be restored when the computer restarts. However, when the computer restarts, Windows loads the existing drivers before it restores the later versions of the drivers. Because the driver versions do not match the versions of the restored catalog files, the restart process stops.

[What?!?]

Workaround

To recover from the failed restart

After the failure occurs, you should be able to restart the computer and then enter the Windows Recovery Environment (WinRE). To do this, you may have to use a hardware restart switch, and you may have to restart two times.

Miscellany

Yesterday… Edge for macOS was officially released:
https://www.microsoftedgeinsider.com/en-us/download/
https://www.microsoftedgeinsider.com/en-us/download/?platform=macos
If someone just cannot wait for Edge on MacOS (I’m unsure who that might be) the least stable “Canary Channel” version of Edge for macOS is now available for download.

SQRL presentation tomorrow during May OWASP Los Angeles Monthly Dinner meeting:
https://www.meetup.com/OWASP-Los-Angeles/events/259755502/

Before last week's podcast I asked the organizer whether it would be okay for me to mention it on the podcast so that some of our listeners in the Los Angeles area might come and say Hi. Since then the confirmed attendance jumped from 51 to 76… so it's tomorrow after work and we're going to have a ball talking about, demonstrating and answering everyone's questions about SQRL.

And speaking of SQRL, I encountered this question from someone who wanted to play with it:

SQRL

Hi Steve: Since, in this area, I am not the quickest squirrel in the park, I'm experiencing a few issues. When for example I try to log in to Facebook using SQRL, I have no clue how to do that. I may be rather naive, but I kinda had the idea that SQRL would "do the dirty work" for me, but it seems not to be the case, at least not in Facebook.

I have installed SQRL. It's visible both as a Firefox add-on and on the bottom bar (right) where it's supposed to be.

I am not a complete fool when it comes down to geek stuff, but I am not the 100% total geek either. When I have started my laptop, opened Firefox, and wish to login at some site, shouldn't sqrl suggest how to do that with the help of sqrl? One of the things that don't work for me is the command code I should enter, in order to create the xpi file. This just didn't happen. I have no clue as to what an xpi file is, and I'm afraid there is no such file on my laptop. I do have a file named sqrl@pass.dog.xpi or the like. Is that it?

Could someone please make a step-by step explanation on how to actually use sqrl, when it doesn't suggest a SQRL login by itself? Am I expecting miracles, or am I a little slow here?

To be honest, I have been wishing for something like this for years, and love Steves other stuff to bits, having followed GRC for years and years, so I posted this in the hope of getting help before I go nuts.

Many kind regards

Thanks for your note and for your interest. SQRL is not like another "password manager." If it were, it would not actually solve any of these traditional problems. For SQRL to work it =requires= support from the website being logged into. So, for example, FaceBook does not yet support SQRL. (For that matter, nowhere but the SQRL forums and a handful of test and demo servers do yet.)

So this is going to take a while. We all know that change of this kind never happens quickly. But it could never happen at all if a truly workable solution was not on offer and available to be kicked and tried, wrung out, tested and evaluated. That's where we are today.

No one knows whether this WILL happen. But after so many years of half-baked solutions it became clear that it wasn't ever GOING to happen unless a good solution was first at least made available to try. Today, as everyone will soon see and be able to play with, we have that.

One of the things that has occurred to me as I've been evangelizing about this a bit is that by asking for a bit of SQRL support from websites, SQRL also succeeds in unifying the sign-on experience across the Internet:

- You see that a site supports SQRL.
- You click on "Sign me in with SQRL"
- Your locally installed SQRL client pops up to ask for your permission to sign into that website for you... and you're in.
- And it works exactly the same way everywhere.

CPU.fail

...or… Speeding Up is Hard to Do."

Teams from the Graz University of Technology, KU Luven and Cyberus Technology have been at it again. And if we ever wanted a more perfect example of Bruce Schneier's sage observation that attacks never get weaker, they only ever get stronger, we have it here. What these guys have done, and have been sitting (probably impatiently) on for the past year is a truly significant advancement to practicality of the mostly theoretical attacks on speculation and micro-architectural performance boosting that we began talking about right from the start of last year.

So what we have now, after more than a year of trouble with mostly-Intel processors, is the generalization of these problems, describing them as Microsoft terms it in their most recent advisory: "Microarchitectural Data Sampling vulnerabilities."

To the original work which first brought us Meltdown, the updated https://CPU.fail site adds ZombieLoad, RIDL, FALLOUT and Store-To-Leak Forwarding. Each which is a different exploitation of subtle features of Intel processors produced over the past eight years which make is extremely difficult for it to keep secrets.

The ZombieLoad attack resurrects your private browsing-history and other sensitive data. It allows to leak information from other applications, the operating system, virtual machines in the cloud and trusted execution environments.

The RIDL attack allows to leak information across various security domains from different buffers, such as line-fill buffers and load ports, inside Intel processors. RIDL demonstrates attacks on other applications, the operating system, other virtual machines and trusted execution environments.

The Fallout attack allows to read data that the operating system recently wrote and to figure out the memory position of the operating system, thus strengthening other attacks. (Defeats KASLR)

Store-To-Leak Forwarding exploits CPU optimizations introduced by the store buffer to break address randomization, monitor the operating system or to leak data when combined with Spectre gadgets.

For Microsoft's part...

On May 14, 2019, Intel published information about a new subclass of speculative execution side channel vulnerabilities known as Microarchitectural Data Sampling.

An attacker who successfully exploited these vulnerabilities may be able to read privileged data across trust boundaries. In shared resource environments (such as exists in some cloud services configurations), these vulnerabilities could allow one virtual machine to improperly access information from another. In non-browsing scenarios on standalone systems, an attacker would need prior access to the system or an ability to run a specially crafted application on the target system to leverage these vulnerabilities.

These vulnerabilities are known as:

CVE-2018-12126 - Microarchitectural Store Buffer Data Sampling (MSBDS)?
CVE-2018-12130 - Microarchitectural Fill Buffer Data Sampling (MFBDS)
CVE-2018-12127 - Microarchitectural Load Port Data Sampling (MLPDS)
CVE-2019-11091 - Microarchitectural Data Sampling Uncacheable Memory (MDSUM)

From the CPU.fail site:

Q: Am I affect by this bug?
A: Most certainly, yes.

Q: Are these software bugs?
A: No. These are bugs in the processor. Software can work around these bugs, which costs performance. Future processors will have integrated fixes.

Q: Can I detect whether someone has exploited this leakage against me?
A: We do not have any data on this. The exploitation might not leave any traces in traditional log files.

Q: Can my antivirus detect or block these attacks?
A: While possible in theory, this is unlikely in practice. These attacks are hard to distinguish from regular benign applications. However, your antivirus may detect malware which uses the attacks by comparing binaries after they become known.

Q: Has this been abused in the wild?
A: We don't know.

So, now, back to the researchers...

ABSTRACT
In early 2018, Meltdown first showed how to read arbitrary kernel memory from user space by exploiting side-effects from transient instructions. While this attack has been mitigated through stronger isolation boundaries between user and kernel space, Meltdown inspired an entirely new class of fault-driven transient execution attacks. Particularly, over the past year, Meltdown-type attacks have been extended to not only leak data from the L1 cache but also from various other microarchitectural structures, including the FPU register file and store buffer.

In this paper, we present the ZombieLoad attack which uncovers a novel Meltdown-type effect in the processor's previously unexplored fill-buffer logic. Our analysis shows that faulting load instructions (i.e., loads that have to be re-issued for either architectural or microarchitectural reasons) may transiently dereference unauthorized destinations previously brought into the fill buffer by the current or a sibling logical CPU. Hence, we report data leakage of recently loaded stale values across logical cores. We demonstrate ZombieLoad's effectiveness in a multitude of practical attack scenarios across CPU privilege rings, OS processes, virtual machines, and SGX enclaves. We discuss both short and long-term mitigation approaches and arrive at the conclusion that disabling hyperthreading is the only possible workaround to prevent this extremely powerful attack on current processors.

[...]

CONCLUSION:

With ZombieLoad, we showed a novel Meltdown-type attack targeting the processor's fill-buffer logic. ZombieLoad enables an attacker to leak recently loaded values used by the current or sibling logical CPU. We show that ZombieLoad allows leaking across user-space processes, CPU protection rings, virtual machines, and SGX enclaves. We demonstrated the immense attack potential by monitoring browser behaviour, extracting AES keys, establishing cross-VM covert channels or recovering SGX sealing keys. Finally, we conclude that disabling hyperthreading is the only possible workaround to mitigate ZombieLoad on current processors.

So what does this mean for our listeners?

~30~