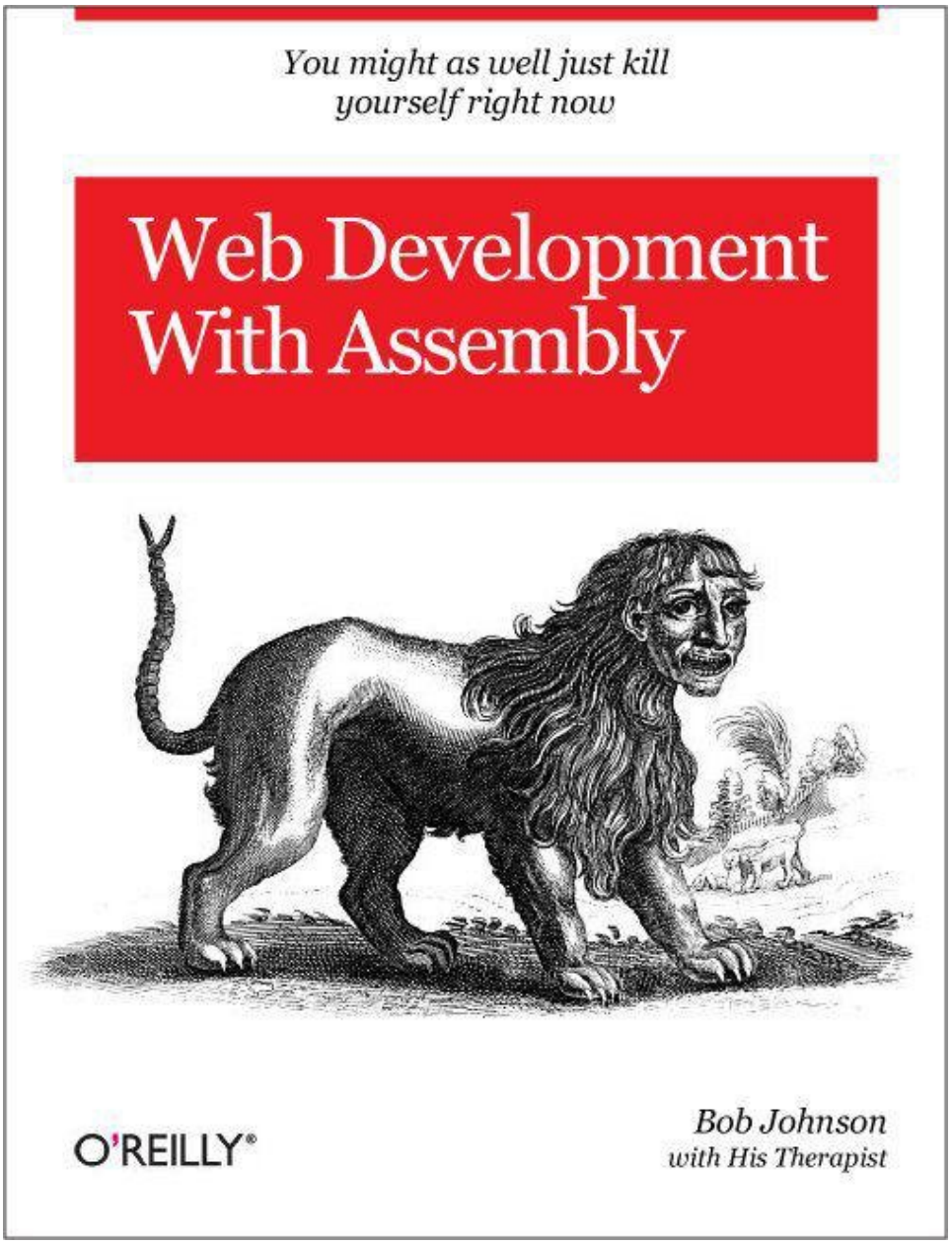
Security Now! #840 - 10-12-21

0-Day Angst

This week on Security Now!

This week we look at Microsoft's decision to finally disable Excel's legacy XLM by default, but not for everyone. We look at Google's warning sent to more than 14,000 of its Gmail users and at their move toward enforced two-step verification. We look at recent hacking and ransom payment legislation and at last week's massive breach at Twitch. We cover the emergency Apache web server update and the mass exodus from WhatsApp during last week's Facebook outage. We look at new Windows 11 side effects and at Patch Tuesday. We close the loop with some listeners and I quickly update on SpinRite's progress. Then we settle down to consider the true significance and import of the various year-to-date 0-day counts.



Windows 11 Watch

"AllowUpgradesWithUnsupportedTPMOrCPU"

That's the name of a registry key whose purpose is made quite clear by its name. And it does, indeed, do what we would think.

1. At the HKEY_LOCAL_MACHINE\SYSTEM\Setup\MoSetup Registry key (which already exists),
2. Create a "REG_DWORD" value named "AllowUpgradesWithUnsupportedTPMOrCPU",
3. Set it to "1" and reboot for the change to take effect.

With that registry value set, the Windows 11 setup will upgrade even over a TPM 1.2 (you do still need at least v1.2 with this value set) and without being blocked by a perfectly fine CPU version.

Simon Zerafa / @SimonZerafa
Replying to @SGgrc: Yes but not receiving security updates will be an issue. Probably best to avoid Windows 11 until the dust settles ???????

First of all, I agree that Win11 is best avoided (we'll be looking at a few additional reasons why in a moment). But in reply to Simon's note about security updates, which Microsoft now says such systems ***may*** not receive, I tweeted:

Simon: "The Win11 requirements were always complete B.S. They tried to make a power play. It failed. And just as that collapsed, there's no reason to think that such systems will not actually receive all security updates. They just want to keep pushing back. But they lost this round."

And really, what possible value can it be to Microsoft, after creating an explicit registry entry called "AllowUpgradesWithUnsupportedTPMOrCPU", to then stubbornly refuse to provide those machines with security updates? That would be nuts. Today, being Patch Tuesday, might be too soon to know whether those systems which were upgraded over Microsoft's decreasingly strong objections will receive updates. But we'll certainly know next month since by then, many unqualifying Win10 machines will have been gently nudged over to Windows 11. Stay tuned.

Bruno Zuber @bzu *(Joined December 2008)*
Steve, 2015: Never10
Steve, 2021: How to upgrade to Win 11 even if your hardware is not supported!

Touché! <g> I use Win7 through every morning and early afternoon at my primary workspace. I'm sitting in front of it right now. Though my microphone and earphones are connected to a tiny little Win10 machine which I used for connecting to TWiT. And I sit in front of Win10 on that Intel NUC with the wide curved screen every evening. Furthermore, any new machine I setup would be Windows 10 and now, I suppose, yes... Windows 11. Microsoft's going to do what Microsoft's going to do. I harbor no illusions about anyone's ability to affect their actions.

Fussing around with an operating system is an entirely reasonable preoccupation. I've spent or wasted untold hours fussing with Windows through the decades. But Simon's also right that depending upon how much time you enjoy spending fighting with the system's default setting, and with the possibility of new incompatibilities, it probably would be far saner to wait a bit. And speaking of new Windows 11 incompatibilities…

Joel, over at ExtremeTech wrote: *"If you're contemplating upgrading to Windows 11 on an AMD system, you may want to hold off just a bit. The semiconductor design firm (AMD) has confirmed that Windows 11 performance is a bit lower on Ryzen CPUs than under Windows 10 right now."*

AMD processors running some apps up to 15% slower
https://www.amd.com/en/support/kb/faq/pa-400

AMD has posted a notice that even on their 175 different processors where Microsoft says Windows 11 installs and runs great, due to unspecified trouble with AMD's L3 caching "Measured and functional L3 cache latency may increase by ~3X."

Under the "Impact" of this, their notice said that:

- Applications sensitive to memory subsystem access time may be impacted.
- Expected performance impact of 3-5% in affected applications, 10-15% outliers possible in games commonly used for eSports.

In addition to that, AMD said that "UEFI CPPC2 (which is the support for their so-called "preferred core") may not preferentially schedule threads on a processor's fastest core." I wasn't aware that not all cores are created the same. Intels are symmetrical. But apparently not all of AMD's are.

To address both of these issues, AMD has said that updates to Windows and software are in development to address this issue with expected availability in October of 2021.

Over at ExtremeTech, Joel concluded his treatment of this by noting:

> *There's no firm timeline on when fixes will be available for either bug, but AMD is promising they'll be ready this month. This issue is separate from the performance-impacting security features that are baked into Windows 11, which impact AMD Zen performance by 4-5 percent if left enabled on an OEM system or turned on by an enthusiast. While these two issues are unrelated, the net effect of them knocks most of a generation's worth of improvement off AMD's CPU cores. Enthusiasts will want to be careful when using OEM PCs, both as far as driver updates and underlying security configurations.*
>
> *At the same time, it's not unusual for a brand-new OS to have some teething problems, so this isn't likely to represent some kind of long-term referendum on Windows 11's gaming performance. As we've covered, Windows 11 is a bit more 'meh' than some of Microsoft's previous releases. Gamers don't need to be in any hurry to jump for the new OS, and while there's no reason to specifically downgrade to Windows 10, there's no great reason to upgrade to Windows 11, either.*

So, thanks, Joel, for the AMD gamer's perspective on Win11. Also...

The Windows 10 taskbar on Windows 11
New Windows 11 users are reporting that after upgrading over Windows 10, their Windows 11 retained the old start menu and that there was no Windows store. An active thread on Reddit suggests re-running a full Windows 11 reinstall, or creating a new user and deleting the old user.

It appears that something about user profiles are getting messed up. So abandoning the old profile that was imported from Windows 10 and restarting with a new profile under Windows 11 appears to resolve the trouble. Unfortunately, all prior user settings and some apps may also then need to be restored and reinstalled. The earlier releases also had this behavior, so it's been seen before by those who enjoy playing with their environment for its own sake. And, again, no idea when this will be resolved.

Microsoft is disagreeing... with themselves
When performing the upgrade, some users are stopped with the message "This PC doesn't currently meet all the system requirements for Windows 11" even when their hardware is compatible. And what makes this more confounding is that when those users run the latest PC Health Check app, they're told that their hardware is compatible and will work without trouble with Windows 11.

We have an update on the Windows Explorer RAM leak I mentioned previously...
As we know, ever since the release of the Win11 previews, File Explorer has been experiencing a memory leak causing the application to use too much system memory. For some, the leak has caused file explorer to use 1GB of memory after opening several folders. And, as I previously noted, after File Explorer is closed, that memory is not released back to the system. It remains unavailable until Windows 11 is restarted. The good news is that Microsoft found and fixed the issue in Win11 22454 preview build for the Insider 'Dev' channel. It's not known when it will be widely public.

VirtualBox and Windows HyperVisors don't get along
At the moment, Hyper-V, Windows Hypervisor and VirtualBox are mutually incompatible. So Windows 11 setup won't agree to setup until they are removed. Oracle and Microsoft are working to get that fixed.

Dropped UDP packets with network optimization
Intel produces something called "Killer" and Dell calls their similar offering "SmartByte." Both of these attempts to optimize network throughput by prioritizing network packet flow is a sort of automatic Quality of Service (QoS) system. The only trouble is, for some reason, when they are prioritizing UDP packets, those packets get dropped, lost and forgotten under Windows 11. This problem is on Microsoft's list of known problems with Windows 11 and it's expected to be fixed in today's Patch Tuesday.

Patch Tuesday

And as I mentioned above, today is Patch Tuesday. We're entering into an era where we're going to be watching two versions of Windows being fixed on the fly. Apparently, today there will already be patches for the Windows 11 released last week. Since nothing really changes under the covers from one Windows release to the next — which is why most of the troubles found once affected Windows 7, 8.1 and 10. Now, with Windows 7 only still being supported for organizations who pay, we'll have updates which broadly affect 8.1, 10 and 11, with a bunch of extras for Windows 11 only due to the fact that Microsoft clearly shipped Windows 11 well before it was ready for release. So next week we'll learn what today and the rest of this week has wrought for Windows.

Security News

The Joy of the (new!) Default: Excel 4.0 macros to be disabled.

For years, Excel 4.0 macros, known as XLM's, have been one of the most abused features of Microsoft's Office suite. Excel Macros have been a favorite of malicious campaigns including TrickBot, Qbot, Dridex, Zloader, and many more.

Excel Macros were introduced nearly three decades ago, back in 1992 with the release of Excel 4.0. These macros provide users with a means of executing commands from within Excel cells. Yeah, you know, what could possibly go wrong? And although these XLM-style macros were superseded with the release of Excel 5 which introduced VBA (Visual Basic for Applications)-based macros, naturally, to be backward compatible, support for XLM macros has remained in place to this day.

Talking out of both sides of their mouth, due to the continued known abuse of XLM macros, while they have deliberately left XLM macros enabled, Microsoft has been recommending that users switch away from and disable this style of macro for years in favor of their newer and more secure VBA macros. VBA macros actually do have the opportunity, at least, to be more secure since VBA macros support the Antimalware Scan Interface (AMSI), which can be used by security software to scan macros for malicious behavior. After many many years of dragging their feet, Microsoft just added support for AMSI scanning to XLM macros in March. But this was seen as much too little and much too late.

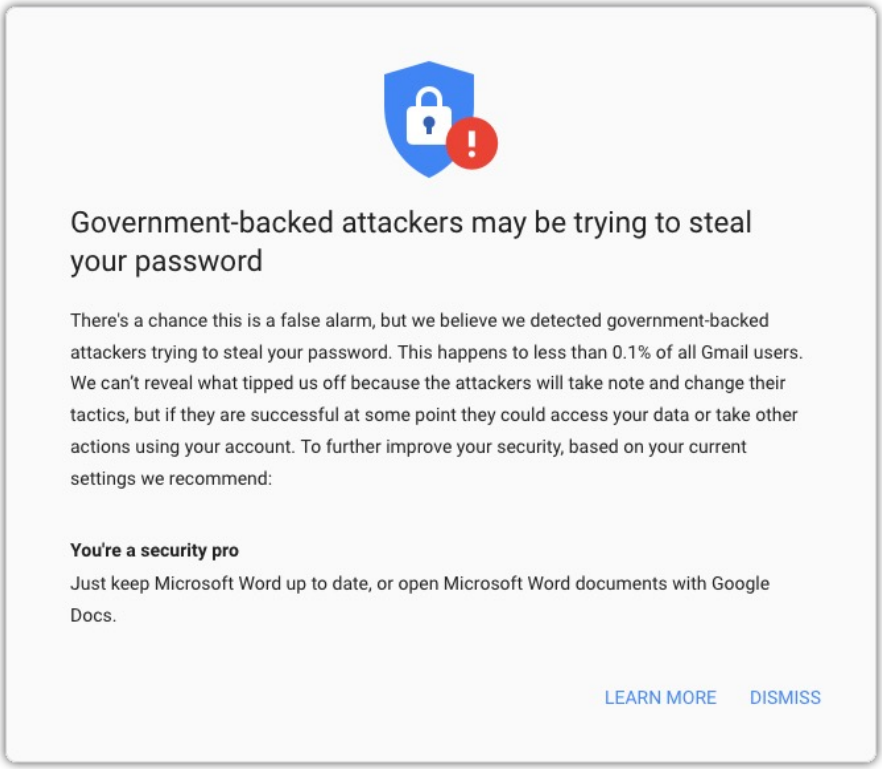
Microsoft may have finally been spurred to (some) action because of a huge relatively recent spike in XLM abuse which was first noted in early 2020. A number of security researchers noted the sudden and unexplainable increased attention that XLM macros had been getting from numerous top-tier bad guys. Reports from VMWare, ReversingLabs, Lastline, MadLabs, Expel, DeepInstinct, and others observed a spike in malware strains and threat actors abusing XLM macros for anything from cyber-espionage to banking trojans, ransomware, and cryptocurrency theft.

Finally, this past summer, security researchers began loudly and publicly criticizing Microsoft for leaving users exposed to attacks, asking for more action from the Gods of Redmond... namely that XLM macros — long having been of only legacy value — should be disabled by default within Office applications. In this way, the researchers have argued, the companies which actually still rely upon legacy XLM could selectively re-enable it for their employees, while everyone else who is being actively abused by having XLM always enabled would then be, and remain, protected from Excel documents containing malicious XLMs.

The logic, of course, is solid and flawless. But, believe it or not, Microsoft will still not be disabling this massively abused 30 year old technology by default for everyone. They will, however, be disabling it for their paying subscribers, as part of the Microsoft 365 service.

Any enterprise admins listening to this podcast should know that Windows group policies can be used to disable unneeded and unused XLM macro capabilities to protect an enterprise's users. And non-enterprise end users can do this for themselves by opening Excel and going to “Excel Options”, “Trust Center”, then click on the “Trust Center Settings...” button and select “Macro Settings” on the left.

Google warns Gmail users of phishing attempts
Google recently sent warning notices to more than 14 thousand users of Gmail warning that <quote> "Government-backed attackers may be trying to steal your password..."



These notices were sent to notify Gmail users that they've been the target of a spear-phishing attack orchestrated by a state-sponsored hacking group.

I'm familiar with the term "Apex Predator" and the idea sort of gives me chills. Wikipedia says: "An apex predator, also known as an alpha predator or top predator, is a predator at the top of a food chain, without natural predators." But until now I've never encountered the term in the context of the security industry. But we now have what are being called "Apex Threat Actors" and their tracking and identification is the mission of Google's TAG team. Google's Threat Analysis Group is led by Shane Huntley. We've been mentioning the TAG team often, recently, since they have been locating and reporting many very valuable vulnerabilities In everyone's software.

Shane told a reporter for "The Record" that "In late September, we detected an APT28 phishing campaign targeting a large volume of Gmail users (approx 14,000) across a wide variety of industries. This particular campaign comprised 86% of the batch of warnings we sent for this month. These warnings indicate targeting, NOT compromise. If we are warning you there's a very high chance [that] we blocked [the attempted attack]." And, indeed, in this case all attempts were blocked.

Huntley added that "If you are an activist/journalist/government official or work in national security, this warning shouldn't be a surprise. At some point, some government-backed entity will probably try to send you something."

The APT28 group is known by many names including Fancy Bear, which both the FBI and NSA directly link to Russia's military intelligence apparatus—and in particular to the Russian General Staff Main Intelligence Directorate (GRU) 85th Main Special Service Center (GTsSS), military unit 26165.

The APT28 / Fancy Bear name comes up often because they've been one of the most active threat actors over the past decade, and the group has often relied on spear-phishing emails in pursuit of targets of interest. Their aim is to breach inboxes, get access to sensitive documents and communications, and then pivot to other individuals or internal networks.

Anyone receiving one of those eMail warnings, or anyone who might be a high-value target, a journalist, politician, celebrity, or CEO, is strongly advised to consider enrolling in Google's Advanced Protection Program for work and personal emails." The Advanced Threat Protection adds and activates additional security protections for high-risk accounts.

And although we're talking about this particular instance, such warnings are not a new Gmail feature. Google has been sending alerts about attacks carried out by state-sponsored entities since 2012.

Google takes first step toward universal 2SV

Okay. So back on May 6th, Google posted "A simpler and safer future — without passwords" which stated that they were embarking on a campaign to auto-enroll all of their users in two-step verification — which they call 2SV — by default. Note that they call this 2SV as opposed to the industry's 2FA — two-factor authentication term. At the time, I noted that I wished them luck since that was sure to be a heavy lift.

https://blog.google/technology/safety-security/a-simpler-and-safer-future-without-passwords/

Last Tuesday the 5th they explained how the first step of this campaign was going to happen, and that it was underway with their posting: "Making sign-in safer and more convenient":

https://blog.google/technology/safety-security/making-sign-safer-and-more-convenient/

Okay, well at least safer. I would argue that there's nothing more convenient about it if they're only adding steps, as they are, without removing any steps. What makes SQRL truly convenient, by comparison, is that it completely replaces both identification and all authentication with its single step. But in any event...

Google has announced their plans to auto-enroll 150 million user accounts into their two-step verification (2SV) system by the end of the year. It turns out, these are accounts where Google's 2SV login **can** be enabled, but where users have not done so on their own. Google wrote: *"Right now, we are auto-enrolling Google accounts that have the proper backup mechanisms in place to make a seamless transition to 2SV."* By "proper backup mechanisms in place" I assume Google means that when they break something by doing this unilaterally there will be some reasonable recovery path.

And I really do appreciate Google's position on this. **User's won't budge.** They just won't. *"Yeah yeah yeah, everything's fine, don't bother me with whatever you're selling."* But they'll surely squawk loudly if someone sneaks into their Google account and starts mucking around with their life. So, to make this happen, Google is going to need to be proactive, for their users' own good... even if Google needs to get all up in their face.

Google's posting explains that this will apply to users with modern smartphones that run recent versions of Android. Once Google proactively and unilaterally enables the 2SV feature, users will be asked to confirm a prompt that appears on their Android smartphone every time they log into their Google account on a new device, app, or browser. That certainly seems reasonable and not at all burdensome. And it would certainly go a long way toward preventing a large class of current remote abuse. And if Google has end-around-access to the registered smartphones running their Android OS, enabling, sending and receiving a realtime push notification shouldn't be a big problem.

So, today's announcement is the first step in Google's ambitious plan to enable 2SV login support for **all** of its users by default. This one is just the start. As part of Google's long-term goal, more users will have 2SV enabled on their accounts going forward as part of a carefully executed staggered rollout plan in order to avoid large breakage. I still think this is going to be an ambitious and heavy lift. It'll be interesting to see what comes next since this was certainly the most obvious and easiest step for them to take.

The US Senate approves some hacking and ransomware legislation

Last Wednesday, the US Senate's Homeland Security Committee advanced two bills aimed at boosting the U.S. government's insight into cyberattacks on critical infrastructure operators and the private sector, as well as federal agencies.

By voice vote the committee approved the Cyber Incident Reporting Act, which would give critical infrastructure owners and operators up to 72 hours to report hacks and 24 hours to disclose ransom payments.

The Senate Homeland legislation mirrors a bipartisan measure from the House Homeland Security Committee that was attached to the House's annual defense policy bill as an amendment. The fact that the bills in each chamber of Congress are aligned suggests that we're going to get that agreed to and signed into law.

The senate bill also took on ransomware by requiring organizations, including businesses with more than 50 employees, nonprofits, and state and local governments, to notify the CISA if they make a ransom payment.

The committee rejected an amendment that would limit the scope of ransom payment reporting amendment to critical infrastructure operators. Many members voiced concern that the mandate would prove burdensome to smaller businesses. However the lawmakers adopted by voice vote an amendment that would, among other things, exempt religious organizations from having to report ransom payments. And the committee later adopted an amendment which would use the Small Business Act's definition for "small business concerns" to exempt small businesses that meet that definition from having to comply with the ransom payment reporting requirement in

the bill. The definition does not set a single fixed threshold based on the number of employees for all businesses.

So, before long, enterprises which do not meet the Small Business Act's definition of a small business will be required, by law, to report ransom payments made to ransomware operators or their affiliates. And all operators of critical infrastructure will be required to report any and all hacks of their facilities.

Amazon's "Twitch" service was hacked bigtime!
Last Wednesday, we learned that Twitch was majorly hacked. And that they first learned of it when 125 gigabytes of their internal proprietary data appeared in a massive online Torrent anonymously released on 4Chan.

Twitch said that no user passwords or credit card numbers were exposed, but if that's true it was about the only thing that wasn't. They said: *"At this time, we have no indication that login credentials have been exposed. Additionally, since full credit card numbers are not stored by Twitch, full credit card numbers were not exposed."* That does sort of sound as though perhaps they had been keeping the last four digits, which is a common method of allowing a user to select a blinded card number.

Twitch said it reset all stream keys as a result of the incident. So users who stream to Twitch would need to obtain a new one from their Twitch profile backends and the Amazon owned company said that while it is still investigating the breach, it believes the breach occurred due to an *"an error in a Twitch server configuration change that was subsequently accessed by a malicious third party."*

The massive data repository contained...

- Entirety of Twitch.tv, with commit history going back to its early beginnings
- Mobile, desktop and video game console Twitch clients
- Various proprietary SDKs and internal AWS services used by Twitch
- Every other property that Twitch owns including IGDB and CurseForge
- An unreleased Steam competitor from Amazon Game Studios
- Twitch SOC internal red teaming tools (lol)
- And creator payout reports from 2019 until now.

Among the treasure trove of data, the most sensitive folders are the ones containing information about Twitch's user identity and authentication mechanisms, admin management tools, and data from Twitch's internal security team, including white-boarded threat models describing various parts of Twitch's backend infrastructure.

The unknown leaker promised to release more data, claiming that this was only the first batch, but they didn't provide a timeline. The threat actor said they leaked the data as a response to Twitch's poor handling of "hate raids," which are bot attacks that have flooded the chats of top streamers with abusive content. Although part of what was leaked shows that Twitch was getting ready to deal with that trouble.

The source of the leak appears to be an internal Git server whose domain name is **git-aws.internal.justin.tv**. "Justin.tv" was the name of the original company prior to its rebranding as "Twitch". Since this occurred 10 years ago back in 2011, that Git server appears to be part of some very old infrastructure.

The leaker labeled this "part one," suggesting that more data might be forthcoming in the future. The biggest question is why no alarms were triggered, not only as a result of the deep internal compromise, but also during the exfiltration of 125 gigabytes of the organization's highly proprietary data.

A major Apache webserver update introduced a new critical 0-day error

The newly introduced vulnerability was discovered and reported to the Apache team by security researcher Ash Daulton and the cPanel Security Team on Wednesday, September 29, 2021. It was being actively exploited in the wild, so it was a true 0-day and consequently the fix for it came pretty quickly.

It's unclear how long the vulnerability was being exploited, but the Apache group was asked and they side-stepped the question in a written reply, saying:

As Apache HTTP Server 2.4.49 was only released a few weeks ago it's likely many users will not have upgraded yet. If and how this issue can be exploited is highly dependent on how users will have configured the server. If you are using 2.4.49, it is recommended that you upgrade to the latest version instead of using access control configuration as a mitigation. On a default installation, an attacker could still use the flaw to obtain the source code of interpreted files like CGI scripts.

What happened was that the release of Apache HTTP Server version 2.4.49 fixed a slew of security flaws including a validation bypass bug, a NULL pointer dereference, a denial-of-service issue, and a severe Server-Side Request Forgery (SSRF) vulnerability. But the major update also inadvertently introduced a separate, critical issue: a path traversal vulnerability that can be exploited to map and leak files.

The developers wrote: "An attacker could use a path traversal attack to map URLs to files outside the expected document root. If files outside of the document root are not protected by "Require all denied" access control, these requests can succeed. Additionally, this flaw could leak the source of interpreted files like CGI scripts."

Positive Technologies has reproduced the bug and Will Dormann, vulnerability analyst at CERT/CC, says that if the mod-cgi function is enabled and the default "Require all denied" function is missing, then the vulnerability is as RCE [remote code execution] as it gets."

The new trouble only impacts Apache HTTP Server 2.4.49 which was, as I noted, only a few weeks old. Even so, as of last Wednesday, approximately 112,755 Apache servers were running the vulnerable version, with roughly 40% of those residing in the United States.

The group who discovered the trouble reported it privately September 29 and the fix was made available just five days later in version 2.4.50 which appeared on October 4th.

Last Week's Mass Exodus from WhatsApp
During last week's 6-hour Facebook services outage, the alternative Signal and Telegram secure messaging platforms struggled to keep pace with the deluge of new users jumping ship from WhatsApps as they looked for an alternative. Unfortunately, some of those services' new users experienced some lagging service and trouble, since both the Signal and Telegram services struggled to keep their own heads above water amid the roaring new demand. This isn't the first time we've noted that new user sign-up processes might not be scaling as well as they should.

Signal tweeted: *"Signups are way up on Signal (welcome everyone!) Millions of new people have joined Signal today and our messaging and calling have been up and running, but some people aren't seeing all of their contacts appear on Signal. We're working hard to fix this up."*

Since the WhatsApp outage was providing a hard "no" for its use, even services that may have been limping along at times were better than nothing.

Pavel Durov, Telegram's CEO and founder, noted that more than 70 million new users joined Telegram in a single day, following Facebook's outage. He added that this massive deluge of millions of new users led to performance issues as they were all trying to sign up on the messaging platform at the same time. Pavel said: *"The daily growth rate of Telegram exceeded the norm by an order of magnitude, as we welcomed over 70 million refugees from other platforms in one day. I am proud of how our team handled the unprecedented growth because Telegram continued to work flawlessly for the vast majority of our users."*

Closing the Loop

I received a Twitter DM from a listener who asked: *"Steve, I listen every week but a lot is over my head. You would help people like me if you did a short segment on Win vs. MAC. That is to say, since my Windows computer is old and cannot get Win11, instead of buying a new Win computer what about a MAC; trade-offs? Your thoughts. Thanks."*

First of all, all past evidence suggests, and all new evidence confirms, that Microsoft appears to have set the Windows 11 CPU requirements bar quite high. I have a lovely Intel NUC which is by no means old. It runs Windows 10 like greased lightning with a fast 2.6 gigahertz quad core i7-6670HQ with 32 gigs of RAM and TPM 2.0. There's absolutely no reason for that machine not to gleefully run Windows 11. But, so far, Microsoft has said no, which is ridiculous. I expect that this might change once they have pushed as many people as they can up to newer hardware. Once that's done, they'll relax the requirements in the interest of resynchronizing everyone around Windows 11. They'll say something like "We've finally finished performing further testing on older hardware and we've confirmed additional compatibility. So we're further relaxing the requirements. We can now state confidently that if a machine runs Windows 10 it'll be able to run Windows 11." That appears to be true today. But Microsoft wants us to play along for now. We still have four years of Windows 10 support, and four years feels like longer than they'll be willing to wait before they move to reunite everyone under Windows 11, especially in light of things like the presence of the "AllowUpgradesWithUnsupportedTPMOrCPU" registry key.

In my opinion, under no circumstances should you purchase a new computer for the sake of running Windows 11. That lovely Intel NUC I mentioned has a wide screen where it's much more fitting to place the taskbar against the screen's left-hand edge. But at the moment, Windows 11 says no to that.

If a lot of this podcast's content feels like it's a bit over your head, that suggests that you might not be ready for a move to one of the Linux desktop environments which, while certainly discoverable, is still a bit less hand-holding than either Windows or MacOS. So, I would say that remaining right where you are for the next four years of Windows 10 service life, to see whether Microsoft discovers that "what do you know, Windows 11 is so good that it works everywhere Windows 10 does!" happens. I wouldn't be a bit surprised.

Mark James Wilcox / @MarkJWilcox
Just remember the timeline of Windows : 3.1 good, 95 bad. 98 good, Millennium bad. XP good, Vista bad. 7 good, 8 and 8.1 bad. 10 good, 11 … about to follow the pattern? No thanks.

Paint Shop Pro v6
Philip Le Riche / @pleriche
"@SGgrc (SN839) I don't believe it! I thought I was the only person on the planet still using PSP6! Simple, does 95% of common tasks. (I'd use the Gimp but learning curve too steep for occasional use.)"

I replied with a shorter version of: Yep! Paint Shop Pro v6. THE best, cleanest, and most straightforward bitmapped graphics editor for Windows ever. I also have PSP7, which was the last one before Corel bought it and ruined it. But I didn't like what JASC did to PSP7... so I returned to v6. I use a couple of plugins, "EyeCandy" and another which does more highly optimized image saving. Any images that appear anywhere on GRC were tailored, trimmed and produced by PSPv6. NOTHING beats it! :) And I'm glad to know that I'm in good company!

Sci-Fi

Just a bit of errata from last week's statement that Apple's new "Invasion" series would be starting **last** Friday. Make that Friday after next, the 22nd, as I originally said the first time I mentioned it. And I haven't yet watched the 4th installment of "Foundation." I'm unsure whether I'll bother.

SpinRite

Last week I shared my decision to rework and rewrite SpinRite's benchmarking technology. Today, I very nearly have all of that work done... and I'm SO glad that I decided to go this route. The new UI display is designed and I'm rolling along very nicely with the implementation of the new system. Rooting out all of the old code was very gratifying, since it embodied a number of kludges which had worn out their welcome long ago. But they were required, as I explained last week, until the use of the flaky old counter/timer could be reliably retired and replaced by using

the clean and solid up-counter clock which we've had now for some time. I'm working on the code to dynamically display the ratio of accumulated bytes transferred to accumulated total benchmark time. Thus, as the benchmark runs, both baselines extend and their ratio will settle into the result. Since the drive speeds it will be encountering will run from the very old to the solid state, that code needs to dynamically scale to handle bytes, kilobytes, megabytes, gigabytes and terabytes per second, while being careful to do the math in a sequence that preserves the greatest number of significant digits. I expect to have it nailed down and tested in another day or two, after which I'll release it to the gang in the GRC newsgroups to pound on. I'm excited about that, since this will be the first widespread testing of SpinRite's new IO abstraction system.

0-Day Angst

I was originally planning to lead with this topic, under the podcast's "0-Day Watch" heading. But as I was exploring the conclusions that follow from the continual march of 0-days we've been tracking this year, it developed into something more. And since no other topic had risen to the level of title, I decided to move this discussion to the end, promoting it to today's title topic.

So, it's Apple's turn (again)
Apple has just patched iPhone 0-day in iOS 15 and it's an authentic 0-day because it is being exploited in the wild. Tracked as CVE-2021-30883, the 0-day resides in the IOMobileFramebuffer which is a kernel extension that allows app developers to manage a device's screen framebuffer memory. As a consequence of this flaw, Malicious applications were able to execute arbitrary code with kernel privileges using this vulnerability. And as we know, running one's own malicious code with kernel privileges gives an attacker full control over the device.

As always, Apple is mum about the technical details of the vulnerability or about how the vulnerability was being leveraged. However, an unrelated security researcher immediately posted a detailed technical teardown on GitHub based upon his discovery from comparing the pre- and post-patched code: https://saaramar.github.io/IOMFB_integer_overflow_poc/

As we've noted before, "bindiff'ing", as it's called, is the practice of comparing a compiled binary file containing a now-known vulnerability to the same compiled binary file after it has been patched and repaired. BinDiff is short for "Binary Difference." The researcher wrote:

> *In the last iOS security update (15.0.2) Apple fixed a vulnerability in IOMobileFrameBuffer/ AppleCLCD, which they specified was exploited in the wild (CVE-2021-30883). This attack surface is highly interesting because it's accessible from the app sandbox (so it's great for jailbreaks) and many other processes, making it a good candidate for LPE exploits in chains (WebContent, etc.).* [By LPEs he means Local Privilege Elevations.]
>
> *Therefore, I decided to take a quick look, bindiff the patch, and identify the root cause of the bug. After bindiffing and reversing, I saw that the bug is great, and I decided to write this*

short blogpost, which I hope you'll find helpful. I really want to publish my bindiff findings as close to the patch release as possible, so there will be no full exploit here; However, I did manage to build a really nice and stable POC that results in a great panic at the end :)

Sorry in advance for any English mistakes, I prioritized time over grammar (good thing we have automatic spell checkers:P).

He then proceeds with a very long and satisfyingly detailed breakdown of the now-fixed, or for those not yet patched, soon-to-be-fixed update. When I checked my iPhone, it was still back on the last v14 release. So I updated it to v15.0.2.

Since we now have a "0-Day Watch" section of this podcast, and since we've been counting the Chrome/Chromium 0-days year-to-date, I think it's only fair to note that this latest 0-day in iOS brings Apple's year-to-date count to 17:

CVE-2021	Patch date	Description
1782	February 1	A zero-day impacting macOS, iOS, iPadOS, watchOS & tvOS kernels.
1870	February 1	WebKit zero-day impacting macOS, iOS, iPadOS, and watchOS
1871	February 1	WebKit zero-day impacting macOS, iOS, iPadOS, and watchOS
1879	March 26	WebKit bug impacting both old & new iOS, iPadOS & watchOS
30657	April 26	macOS Gatekeeper bypass abused by Shlayer malware
30661	April 26	WebKit zero-day impacting old & new iOS, iPadOS, watchOS & tvOS.
30663	May 3	WebKit zero-day impacting macOS, iOS, iPadOS, and watchOS
30665	May 3	WebKit zero-day impacting macOS, iOS, iPadOS, and watchOS
30666	May 3	WebKit zero-day impacting macOS, iOS, iPadOS, and watchOS
30713	May 24	macOS TCC bypass abused by XCSSET malware
30761	June 14	WebKit zero-day impacting old-gen iOS devices
30762	June 14	WebKit zero-day impacting old-gen iOS devices
30807	July 26	IOMobileFramebuffer zero-day impacting macOS, iOS, and iPadOS
30858	Sept 13	WebKit zero-day impacting macOS, iOS, iPadOS, and watchOS
30860	Sept 13	0-day in CoreGraphics impacting macOS, iOS, iPadOS, and watchOS
30869	Sept 23	XNU kernel component zero-day impacting iOS and macOS
30883	October 11	IOMobileFramebuffer zero-day impacting iOS and iPadOS

Given the big target that every web browser presents, and the fact that the other 0-day march we've been following is the world's leading web browser, Chrome, it should not come as any surprise that 10 of those 17 0-days were discovered and fixed in Apple's Webkit browser component.

One point of interest was brought out by this researcher, who noted that the July 26th 0-day, CVE-2021-30807, was also an IOMobileFramebuffer 0-day impacting iOS, iPadOS, and macOS. One wonders whether, once Apple removed that earlier 0-day in the frame buffer module, those attackers may have simply switched to using another 0-day they already knew of, and had at the ready, in the same module?

In any event, stepping back from all this a bit, I think it's clear that the engineering practices surrounding the creation and maintenance of the best-designed software, which is what Apple and Google are both capable of producing—and do produce where it matters most—has become so secure that the world's users should feel completely safe using it. But at the same time, that software has become so complex that our current development methodologies, languages and toolsets clearly fall short of creating perfect software.

Perfect software doesn't support a running total of "0-days so far this year," and those tiny imperfections give the likes of Israel's NSO Group, with their Pegasus smartphone spyware, just enough of the toehold to enable highly targeted attacks against the world's highest value targets.

At this stage in the development of the Internet and the use of personal connected devices—predominantly smartphones—what Google has created and accomplished with Android has been phenomenally valuable to all mankind. And placing sharing first, doing it all in plain sight for the world to see and benefit from, to examine and unfortunately to attack, whether it's Android or Chromium, is significantly more difficult than working to keep everything a tightly locked-down secret.

At the same time, if one's goal, rather than altruism, is profit, secrecy has its place.

Apple, whose products are as proprietary as they're capable of being, made a serious strategic security blunder when they chose to share code between iOS and macOS. Sure, it made for huge economies of development. Why continually reinvent the wheel on separate platforms? But the security of iOS is arguably **far** more critical than that of macOS. Yet by merging and sharing their codebases iOS's security has been reduced to the lowest common denominator.

In that list of "Apple 0-days so far this year", I highlighted in red those shared by macOS and iOS. Of the 17 total, 10 were 0-days present on **both** OS platforms. Want to place a bet where those 0-days were first discovered? After the merging of Apple's codebases we've encountered many examples where researchers and attackers were able to reverse engineer the code first on the unprotectable and far more accessible macOS platform (and, by the way, where they really couldn't care less about a vulnerability on macOS), then pivot with what they learned there, to the far more valuable iOS platform, knowing that the same vulnerabilities, though unseen, were likely to exist there too.

The advantage Apple had, and unfortunately squandered, was that any code running on an iDevice is far more easily protected, hidden and kept secret. The favorite slogan "security through obscurity is not security" makes for a catchy phrase. But it's not quite true. Some obscurity, any obscurity, is better than none. And at some point, sufficient obscurity becomes secrecy. "Security through secrecy" does provide true security. After all, we all keep passwords and private keys secret, which, notwithstanding other errors in their management, keeps them secure. Apple's iDevices today are measurably less secure than those of the past because keeping their code secret had true security value.

Although Google's device security does suffer from its total openness, the openness of its Google Play Store and its support for sideloading completely uncurated applications, Google's efforts are providing far more benefit to the world at large, than Apple's.

I titled this podcast "0-day Angst" because I wanted to place the issue of 0-day vulnerabilities front and center and in a sober context. Yes it's true that despite everyone's best efforts, 0-days occur, and there's no reason to believe that the near term future will see any change in that. We're not going to reduce the features and complexity of our software, no new bulletproof languages, development, or environment solutions are apparent, and the truth is, 0-days are more of an embarrassment to their publishers than a global security threat. And they are also inherently self-limiting: The more they are used the more likely they are to be discovered and eliminated.

Thanks to a great deal of effort being made by mainstream software publishers, aided by a massive, distributed and growing community of security researchers (and even with some inadvertent help from the bad guys) today's devices, even though we absolutely positively know they still contain known and unknown vulnerabilities, are more than secure enough for nearly everyone to use without worry. Only those very few who are likely to be targeted by nation-state actors have any real reason for concern.