Security Now! #1020 - 04-08-25

Multi-Perspective Issuance Corroboration

This week on Security Now!

Canon printer driver vulnerabilities enable Windows kernel exploitation. Astonishing cyber-security awareness from a household appliance manufacturer. France tries to hook 2.5 million school children with a Phishing test. Wordpress added an abuse prone feature in 2022. Guess what happened? Oracle? Is there something you'd like to tell us? Utah's governor just signed the App Store Accountability Act. Now what? AI bots hungry for new data are DDoSing FOSS projects. No Microsoft Account? No Microsoft Windows 11. Gmail claims it now offers E2EE. It kinda sorta does. Somewhat. A dreaded CVSS 10.0 was discovered in Apache Parquet. A bunch of terrific listener feedback. What's Multi-Perspective Issuance Corroboration and why must all certificate authorities now do it?

Security News

Canon Printer Drivers Flaw Could Let Hackers Run Malicious Code

The Microsoft Offensive Research and Security Engineering (MORSE) team has identified a crucial security vulnerability within a range of Canon printer drivers, threatening users across various sectors. This vulnerability could reportedly allow malicious actors to compromise printing operations and, in severe cases, execute arbitrary code on affected systems.

This has a CVSS score of 9.4, indicating a high-severity risk, due to its low attack complexity, the lack of required privileges or user interaction, and the potential for high-impact compromise of confidentiality. The flaw provides a path to deliberate memory corruption during EMF Recode processing, which malicious applications can leverage to insert and execute their own code.

And, unfortunately, this opens systems that do not use Canon printers to BYOVD attacks there BOYVD abbreviates "Bring Your Own Vulnerable Driver". Since these vulnerable Canon printer drivers are signed by Microsoft, they can be dropped onto any Windows system, where they take up residency in the kernel, and then facilitate attacks.

The affected drivers are used in a variety of Canon printers, including production models, office and home-office multifunction printers, and laser printers. If a malicious application processes a print job through these vulnerable drivers, attackers could gain unauthorized access to execute their own code code that could lead to data breaches or system control.

Canon has acknowledged the issue and has committed to releasing updated printer drivers to address it. Users are strongly advised to keep an eye out for and promptly download and install the latest versions of these drivers to mitigate the risk of exploitation.

What I assume will happen is that endpoint security systems will attempt to detect and block the use of this vulnerability. And once replacement drivers are made available, the vulnerable drivers will no longer be allowed to be downloaded or loaded into the kernel. But even if these measures are taken, all of this introduces a period of vulnerability having many moving parts. No matter what we do, the discovery of vulnerable kernel drivers represents an ever present threat.

And we saw last week just how active the ransomware world has become and remains. Those cretins will leverage any opening they can find.

Fisher Paykel

I was put onto this amazing home appliance company by a piece of feedback email I received from our listener David Morrell. David wrote: *"Every home IoT device maker should follow the lead of this home appliance maker. About the only thing they could have added is advice to use a Yubikey or similar. They really get it. And it even looks like you can buy these New Zealand made home appliances in the U.S.A. Personally, I'm quite happy not having IoT in my home appliances."*

David's note made me curious so I went over to the Fisher Paykel appliance website where I discovered that they have an entire page devoted to the Cyber-Security of their well-connected appliances: https://support.fisherpaykel.com/s/article/Cyber-Security

To give everyone a sense for what's there, they wrote:

> *We are vigilant about securing your connected appliance. We understand that the security of*

our products is of the utmost importance to our customers. We build appliances around these core security values. Security is ingrained in our business culture and in the way we developed your connected appliance. It’s a business policy that security is “built-in” to every aspect of our process. It's built-in during all phases of development, manufacturing, and maintenance. Your appliance is secure without user configuration or specific router settings.

Security by Design: Security controls to protect appliance data, user authentication and authorisation and how the system will be securely maintained are integrated into the functional features of the appliance. The software meets industry best practice coding standards and is developed by the Test-Driven Development software method. Any third-party and open-source software is analysed for security and the safety of your appliance and data. Prior to deployment, the appliance undergoes extensive software security and performance testing. Security penetration testing on the connected system and its components–the appliance, mobile app, and cloud—is done regularly post-deployment. Software updates are released to ensure the appliance has the latest security code to protect your appliance and data.

*Security by Default: Every connected appliance has **all** security features enabled when the appliance is first connected. No special configurations or specific router settings are needed. Your appliance connects to your Wi-Fi router using the WPA3 network security protocol as standard, with WPA2 backwards compatibility. The appliance does this even if your router is not set to this configuration. That's just one example of how Security by Default is engineered into your appliance.*

Defense In Depth: Every component of our connected appliance ecosystem has security controls that provide independent redundancy to protect against malicious attacks. We ensure security controls are implemented in layers for data protection at rest and in transit. This layered approach strengthens the security of our entire ecosystem. We are continuously testing and reviewing the security systems, if needed, these layers can be updated and improved by software updates.

Security by Transparency: Our security controls and methodologies are industry standard. Our goal is to communicate our actions with openness and accountability. We are industry leaders in IoT security and promote transparency to help educate our customers. Reach out to us if you have any questions or concerns. Please see below under our ratings section for current evaluations of our appliance products. We ensure these best practices are applied to your appliance and its IoT ecosystem through regular penetration testing. We work with ethical hackers and security researchers to evaluate the security of your smart appliance and system through third-party evaluations.

Our Ratings: We are proud to have achieved the Gold verification level for UL’s (Underwriter Laboratories) IoT Security Rating. With thorough evaluations, conducted every year since we first achieved this rating, we continually demonstrate Gold Level security capabilities that align with industry best practices. This recognition validates our long-standing commitment to consumer data protection, transparency, and investment in security, and further demonstrates our cybersecurity capabilities to our retailers, and regulators while providing peace of mind to our consumers.

Questions About Security: We are committed to answering your questions or any concerns you may have. With all of our brands, our goal is to ensure your satisfaction, while offering the highest levels of professional service. For security tips on configuring your home router, good security Internet hygiene, and keeping your devices up to date, please read our Smart Home Security Guidance section below.

> *If you suspect that your appliance has been compromised or you have identified a security vulnerability in one of our connected appliances, we urge you to contact our Appliances Security Incident Response Team (SIRT) at is@fisherpaykel.com - Note: we support PGP encryption using the Fisher & Paykel Appliances Information Security PGP Key. We are committed to protecting the privacy and security of their customers. Therefore, we will not disclose, discuss, or confirm any security issue until a full investigation is completed and any necessary press releases, security patches, and releases are made available.*

They then have a section on recommendations for achieving the best security which ends with:

> *Separate networks: Security experts recommend creating separate and secure networks dedicated for your IoT devices that are separate from your network used for banking or e-commerce activities or that which handles your most private and sensitive data. You can further segregate your networks based on the IoT device itself. There are two methods for this when using one Internet connection, (1) using one router and setting up a "guest access" or a "guest network" within the router settings or (2) use separate routers paired with your Internet connection. If you choose to set up a guest network, ensure the password for the guest network is strong and, if available, ensure that access to local network resources is turned off, this may also be called "isolate".*

I'm utterly astonished by these people. And it's a good thing this is April 8th and not last week's April 1st podcast, because this would have made the perfect April Fools Spoof — since no one would ever believe that I hadn't made this entire thing up from scratch.

If the rest of the world designed and built their equipment like this, it feels as though our job here would be done. Wow!

France's Phishing Test – The kids did pretty well.

The French government recently conducted a large-scale phishing test targeting more than 2.5 million middle and high school students. The bait was a link that advertised cheats and cracked games that instead redirected students to a phishing awareness video. According to France's privacy watchdog, over 210,000 students clicked the link, representing roughly one in twelve students. And while 210,000 is a lot of individual students, those young students fared far better than the approximately 1/3rd click rate typically seen in corporate environments.

WordPress added an abuse-prone feature in 2022

As we've observed before, with 521 million websites built on WordPress, which is 43.5% of all websites in the world, Wordpress security is a top concern. So when, three years ago, Wordpress added a feature attackers could only dream of having, it's hardly surprising that it didn't take long for it to be abused. Wordpress's site describes this nifty new feature known as "Must Use Plugins" by writing:

> *Must-use plugins (a.k.a. mu-plugins) are plugins installed in a special directory inside the content folder and which are automatically enabled on all sites in the installation. Must-use plugins do not show in the default list of plugins on the Plugins page of wp-admin (although they do appear in a special Must-Use section) and cannot be disabled except by removing the plugin file from the must-use directory, which is found in wp-content/mu-plugins by default.*

> *For web hosts, mu-plugins are commonly used to add support for host-specific features, especially those where their absence could break the site. Must-use plugins are always on with need to enable via admin and users cannot disable by accident. They are enabled simply by uploading a file to the mu-plugins directory, without having to log-in.*

This, of course, is where cue one of our favorite rhetorical questions: *"**WHAT** could **POSSIBLY** go wrong?"* GoDaddy's Sucuri security team provides the answer to that question and unfortunately it's not rhetorical. To no one's surprise, except I suppose the creators of this very abuse-prone feature, hackers are now abusing this little-known WordPress feature to install and hide their malware from site admins. According to GoDaddy's Sucuri security team, threat actors have been found to be abusing Must Use Plugins since at least February of this year. And that abuse has recently grown worse.

Hackers are breaking into WordPress sites and dropping malware in the mu-plugins folder, knowing it will get automatically executed and won't show up in site backends. As an added benefit, because it's a relatively unknown and under-the-radar feature, many WordPress security tools don't scan the mu-plugins folder for threats. Sucuri has seen attackers use the mu-plugins folder to deploy backdoors and web shells, host SEO spam on hacked sites as well as hijack and redirect traffic to malicious sites. The wide and widening spectrum of abuse suggests this feature is gaining popularity and traction among underground groups. A Sucuri analyst said: *"The fact that we've seen so many infections inside the mu-plugins directory suggests that attackers are actively targeting this directory as a persistent foothold."*

Wordpress site owners and administrators are advised to keep a watch on the content of that folder. If it's currently empty, unused and unneeded, just delete it entirely and make sure it stays deleted. Stepping back from this, it appears that the design of this makes it far too easy to both use and also abuse. With a design like this, it's not possible to have one without the other.

Oracle? Is there something you'd like to tell us? *(Confession is good for the soul.)*
Meanwhile, Oracle, the massive organization with designs on running TikTok and retaining all of TikTok's US domestic data appears to be having a problem with confession. According to Bloomberg sources hackers breached Oracle Health and stole medical data from the company's servers. The hack took place at the end of January, and the hackers are using the stolen data to extort US medical providers. Yet Oracle has said nothing. They've made no report of any breach as is required by law to the US Securities and Exchange Commission.

But wait, there's more. This is the second suspected breach at Oracle after a different hacking group claimed to have hacked the company's Cloud service in early March. Lawrence Abrams, wrote about this for his BleepingComputer site under the headline *"Oracle customers confirm data stolen in alleged cloud breach is valid"* Lawrence wrote:

> *Despite Oracle denying a breach of its Oracle Cloud federated SSO login servers and the theft of account data for 6 million people, BleepingComputer has confirmed with multiple companies that associated data samples shared by the threat actor are valid.*
>
> *Last week, a person named 'rose87168' claimed to have breached Oracle Cloud servers and began selling the alleged authentication data and encrypted passwords of 6 million users. The threat actor also said that stolen SSO and LDAP passwords could be decrypted using the info in the stolen files and offered to share some of the data with anyone who could help recover them.*

> *The threat actor released multiple text files consisting of a database, LDAP data, and a list of 140,621 domains for companies and government agencies that were allegedly impacted by the breach. It should be noted that some of the company domains look like tests, and there are multiple domains per company.*
>
> *In addition to the data, rose87168 shared an Archive.org URL with BleepingComputer for a text file hosted on the "login.us2.oraclecloud.com" server that contained their email address. This file indicates that the threat actor could create files on Oracle's server, indicating an actual breach.*
>
> ***However, Oracle has denied that it suffered a breach of Oracle Cloud and has refused to respond to any further questions about the incident.***
>
> *The company told BleepingComputer: "There has been no breach of Oracle Cloud. The published credentials are not for the Oracle Cloud. No Oracle Cloud customers experienced a breach or lost any data." This denial, however, contradicts findings from BleepingComputer, which received additional samples of the leaked data from the threat actor and contacted the associated companies. Representatives from these companies, all who agreed to confirm the data under the promise of anonymity,* ***confirmed the authenticity of the information.*** *The companies stated that the associated LDAP display names, email addresses, given names, and other identifying information* ***were all correct and belonged to them.***
>
> *The threat actor also shared emails with BleepingComputer, claiming that it was part of an exchange between them and Oracle. One email shows the threat actor contacting Oracle's security email (secalert_us@oracle.com) to report that they hacked the servers.*
>
> *<quote> "I've dug into your cloud dashboard infrastructure and found a massive vulnerability that has handed me full access to info on 6 million users." reads the email seen by BleepingComputer. Another email thread shared with BleepingComputer shows an exchange between the threat actor and someone using a ProtonMail email address who claims to be from Oracle. BleepingComputer has redacted the email address of this other person as we could not verify their identity or the veracity of the email thread.*
>
> *In this email exchange, the threat actor says someone from Oracle using a @proton.me email address told them that* ***"We received your emails. Let's use this email for all communications from now on. Let me know when you get this."***
>
> *Cybersecurity firm Cloudsek has also found an Archive.org URL showing that the "login.us2.oraclecloud.com" server was running Oracle Fusion Middleware 11g as of February 17 of this year, 2025. Oracle has since taken this server offline after news of the alleged breach was reported. This version of Oracle's software was impacted by a vulnerability tracked as CVE-2021-35587 that allowed unauthenticated attackers to compromise Oracle Access Manager. The threat actor claimed that this vulnerability* ***was*** *used in the alleged breach of Oracle's servers.*
>
> *BleepingComputer has emailed Oracle numerous times about this information but has not received any response.*

And, again, in the face of this overwhelming evidence – which arguably borders on proof – Oracle has deliberately chosen to remain entirely silent even though doing so is a clear breach of reporting law.

The U.S. Securities and Exchange Commission (SEC) mandates that public companies adhere to specific reporting requirements following a material cybersecurity incident, such as a database breach affecting U.S. citizens. These requirements, which have been effective since December of 2023, are designed to ensure timely and transparent disclosure of significant cybersecurity events. Specifically, within four business days after discovering that a cybersecurity incident is material, publicly traded companies are required to file a Form 8-K disclosure under Item 1.05. That disclosure must include:

- The nature, scope, and timing of the incident.
- The material impact or reasonably likely material impact on the company's financial condition and results of operations.
- Determination of Materiality: Companies are required to assess the materiality of an incident without unreasonable delay upon discovery.

Oracle knows this. Yet nothing about either of these clearly material major breaches.

Utah’s governor signs the online child safety acts into law.

Meanwhile, I wanted to note that nearly two weeks ago that Utah law we talked about Utah’s legislation passing was signed into law by Utah’s governor, Spencer Cox. Formally known as the “App Store Accountability Act”, or S.B. 142, this new law mostly takes effect a little over one year from now, on May 6th, 2026, when the law’s core requirements, including age verification and parental consent mandates, take effect. This is intended to allow time for app stores, developers, and regulators to prepare for compliance with the new regulations.

As we’ve discussed, this will require Apple and Google’s mobile app stores to verify user ages and require parental permission for those under 18 to use certain apps. This law is the first of its kind in the U.S. and represents a significant shift in how user ages are verified online. The law states that it’s the responsibility of mobile app stores to verify ages — which shifts the onus to Apple and Google and away from individual apps like Instagram, Snapchat and X, to do age checks.

This does beg the question, though, what about apps that are already downloaded and installed from app stores? Are those grandfathered in and allowed to stay without verification measures?

Regardless, the passage of this App Store Accountability Act is expected to trigger the movement of similar legislation from other U.S. states including South Carolina and California. One of the bill’s sponsoring senators said that the new law is designed to protect children, who may not understand apps’ terms of services and, therefore, can’t agree to them. Todd Weiler said: *“For the past decade or longer, Instagram has rated itself as friendly for 12 year olds. It’s not.”*

The Utah law is expected to face legal challenges in fights over its validity but as we know, my own take on this is that something needs to be done. I think that the most recent begrudging proposals being made by Apple and Google make a lot of sense. App store apps need to carry API-readable age appropriate indicators and the devices being used by minors may need to obtain parental permission before inappropriate applications can be downloaded and/or used on age-restricted devices. Problem solved.

Data hungry AI bots are taking down FOSS sites by accident

Okay. It turns out that AI Bots are inadvertently DDoSing FOSS – free and open source software – repositories in their endless quest for more publicly available content to digest.

ArsTechnica did a great job of reporting on a worrisome trend that's been developing and worsening this year. They wrote:

> *Software developer Xe Iaso reached a breaking point earlier this year when aggressive AI crawler traffic from Amazon overwhelmed their Git repository service, repeatedly causing instability and downtime. Despite configuring standard defensive measures—adjusting robots.txt, blocking known crawler user-agents, and filtering suspicious traffic—Iaso found that AI crawlers continued evading all attempts to stop them, spoofing user-agents and cycling through residential IP addresses as proxies.*
>
> *Desperate for a solution, Iaso eventually resorted to moving their server behind a VPN and creating "Anubis," a custom-built proof-of-work challenge system that forces web browsers to solve computational puzzles before accessing the site. Iaso wrote in a blog post titled "A desperate cry for help" that <quote> "It's futile to block AI crawler bots because they lie, change their user agent, use residential IP addresses as proxies, and more. I don't want to have to close off my Gitea server to the public, but I will if I have to."*
>
> *Iaso's story highlights a broader crisis rapidly spreading across the open source community, as what appear to be aggressive AI crawlers increasingly overload community-maintained infrastructure, causing what amounts to persistent distributed denial-of-service (DDoS) attacks on vital public resources. According to a comprehensive recent report from LibreNews, **some open source projects now see as much as 97 percent of their traffic originating from AI companies' bots**, dramatically increasing bandwidth costs, service instability, and burdening already stretched-thin maintainers.*
>
> *Kevin Fenzi, a member of the Fedora Pagure project's sysadmin team, reported on his blog that the project had to block all traffic from Brazil after repeated attempts to mitigate bot traffic failed. GNOME GitLab implemented Iaso's "Anubis" system, requiring browsers to solve computational puzzles before accessing content. GNOME sysadmin Bart Piotrowski shared on Mastodon that only about **3.2 percent of requests (2,690 out of 84,056)** passed their challenge system, suggesting the vast majority of traffic was automated. KDE's GitLab infrastructure was temporarily knocked offline by crawler traffic originating from Alibaba IP ranges, according to LibreNews, citing a KDE Development chat.*
>
> *While Anubis has proven effective at filtering out bot traffic, it comes with drawbacks for legitimate users. When many people access the same link simultaneously—such as when a GitLab link is shared in a chat room—site visitors can face significant delays. Some mobile users have reported waiting up to two minutes for the proof-of-work challenge to complete, according to the news outlet.*
>
> *The situation isn't exactly new. In December, Dennis Schubert, who maintains infrastructure for the Diaspora social network, described the situation as "literally a DDoS on the entire internet" after discovering that AI companies accounted for 70 percent of all web requests to their services.*
>
> *The costs are both technical and financial. The Read the Docs project reported that blocking AI crawlers immediately decreased their traffic by 75 percent, going from 800GB per day to 200GB per day. This change saved the project approximately $1,500 per month in bandwidth costs, according to their blog post "AI crawlers need to be more respectful."*

The situation has created a tough challenge for open source projects, which rely on public collaboration and typically operate with limited resources compared to commercial entities. Many maintainers have reported that AI crawlers deliberately circumvent standard blocking measures, ignoring robots.txt directives, spoofing user agents, and rotating IP addresses to avoid detection.

As LibreNews reported, Martin Owens from the Inkscape project noted on Mastodon that their problems weren't just from "the usual Chinese DDoS from last year, but from a pile of companies that started ignoring our spider conf and started spoofing their browser info." Owens added, "I now have a prodigious block list. If you happen to work for a big company doing AI, you may not get our website anymore."

On Hacker News, commenters in threads about the LibreNews post last week and a post on Iaso's battles in January expressed deep frustration with what they view as AI companies' predatory behavior toward open source infrastructure. While these comments come from forum posts rather than official statements, they represent a common sentiment among developers.

As one Hacker News user put it, AI firms are operating from a position that "goodwill is irrelevant" with their "$100 Billion pile of capital." The discussions depict a battle between smaller AI startups that have worked collaboratively with affected projects and larger corporations that have been unresponsive despite allegedly forcing thousands of dollars in bandwidth costs on open source project maintainers.

Beyond consuming bandwidth, the crawlers often hit expensive endpoints, like git blame and log pages, placing additional strain on already limited resources. Drew DeVault, founder of SourceHut, reported on his blog that the crawlers access "every page of every git log, and every commit in your repository," making the attacks particularly burdensome for code repositories.

The problem extends beyond infrastructure strain. As LibreNews points out, some open source projects began receiving AI-generated bug reports as early as December 2023, first reported by Daniel Stenberg of the Curl project on his blog in a post from January 2024. These reports appear legitimate at first glance but contain fabricated vulnerabilities, wasting valuable developer time.

AI companies have a history of taking without asking. Before the mainstream breakout of AI image generators and ChatGPT attracted attention to the practice in 2022, the machine learning field regularly compiled datasets with little regard to ownership.

While many AI companies engage in web crawling, the sources suggest varying levels of responsibility and impact. Dennis Schubert's analysis of Diaspora's traffic logs showed that approximately one-fourth of its web traffic came from bots with an OpenAI user agent, while Amazon accounted for 15 percent and Anthropic for 4.3 percent.

The crawlers' behavior suggests different possible motivations. Some may be collecting training data to build or refine large language models, while others could be executing real-time searches when users ask AI assistants for information.

The frequency of these crawls is particularly telling. Schubert observed that AI crawlers "don't just crawl a page once and then move on. Oh, no, they come back every 6 hours because lol why not." This pattern suggests ongoing data collection rather than one-time training *exercises, potentially indicating that companies are using these crawls to keep their models' knowledge current.*

Some companies appear more aggressive than others. KDE's sysadmin team reported that crawlers from Alibaba IP ranges were responsible for temporarily knocking their GitLab offline. Meanwhile, Iaso's troubles came from Amazon's crawler. A member of KDE's sysadmin team told LibreNews that Western LLM operators like OpenAI and Anthropic were at least setting proper user agent strings (which theoretically allows websites to block them), while some Chinese AI companies were reportedly more deceptive in their approaches.

It remains unclear why these companies don't adopt more collaborative approaches and, at a minimum, rate-limit their data harvesting runs so they don't overwhelm source websites. Amazon, OpenAI, Anthropic, and Meta did not immediately respond to requests for comment, but we will update this piece if they reply.

In response to these attacks, new defensive tools have emerged to protect websites from unwanted AI crawlers. As Ars reported in January, an anonymous creator identified only as "Aaron" designed a tool called "Nepenthes" to trap crawlers in endless mazes of fake content. Aaron explicitly describes it as "aggressive malware" intended to waste AI companies' resources and potentially poison their training data.

"Any time one of these crawlers pulls from my tarpit, it's resources they've consumed and will have to pay hard cash for," Aaron explained to Ars. "It effectively raises their costs. And seeing how none of them have turned a profit yet, that's a big problem for them."

On Friday, Cloudflare announced "AI Labyrinth," a similar but more commercially polished approach. Unlike Nepenthes, which is designed as an offensive weapon against AI companies, Cloudflare positions its tool as a legitimate security feature to protect website owners from unauthorized scraping.

Cloudflare explained in its announcement: "When we detect unauthorized crawling, rather than blocking the request, we will link to a series of AI-generated pages that are convincing enough to entice a crawler to traverse them." Cloudflare reported that AI crawlers generate over 50 billion requests to their network daily, accounting for nearly 1 percent of all web traffic they process.

The community is also developing collaborative tools to help protect against these crawlers. The "ai.robots.txt" project offers an open list of web crawlers associated with AI companies and provides premade robots.txt files that implement the Robots Exclusion Protocol, as well as .htaccess files that return error pages when detecting AI crawler requests.

As it currently stands, both the rapid growth of AI-generated content overwhelming online spaces and aggressive web-crawling practices by AI firms threaten the sustainability of essential online resources. The current approach taken by some large AI companies—extracting vast amounts of data from open-source projects without clear consent or compensation—risks severely damaging the very digital ecosystem on which these AI models depend.

Responsible data collection may be achievable if AI firms collaborate directly with the affected communities. However, prominent industry players have shown little incentive to adopt more cooperative practices. Without meaningful regulation or self-restraint by AI firms, the arms race between data-hungry bots and those attempting to defend open source infrastructure

> *seems likely to escalate further, potentially deepening the crisis for the digital ecosystem that underpins the modern Internet.*

No Microsoft Account? No Windows 11
If you're attempting to install Windows 11 on a machine using only a local account without signing into Microsoft and you're wondering why doing so appears to have become more difficult or obscure, it could be because Microsoft now intends to make that completely impossible.

In their recent announcement of Windows 11 Insider Preview Build 26200.5516 for the Dev channel, toward the end of a long list of tweaks and changes, filed under "[other]" Microsoft wrote:

We're removing the bypassnro.cmd script from the build to enhance security and user experience of Windows 11. This change ensures that all users exit setup with internet connectivity and a Microsoft Account.

It's unclear to me how forcing either Internet connectivity or being logged into a Microsoft account enhances either a user's security or their experience, but that's what will henceforth be required for all users setting up Windows 11. I don't mean to make a bigger deal out of this than it is. I imagine that anyone setting up Windows 11 will have already made whatever adjustments to their thinking and expectations that may have been required. But it is a change that I wanted to let our listeners know about.

Some of the reporting I saw about this said: *"Microsoft has been trying to force Windows 11 users to install the OS with a Microsoft account for years, but this marks the first time when the company has made it a public policy in one of its blogs."*

And having shared all that, I won't be surprised if there isn't soon a workaround for this, created and popularized by those who hope to retain some independence from Microsoft.

Gmail to get End-to-End Encryption? Huh?
Last week, Google announced and unveiled what they called "end-to-end encryption" for corporate users of Gmail. But boy is it funky. It does encrypt a message in the sender's web browser, where it remains encrypted until it's opened in the recipient's browser, where it's then decrypted. So, technically, yeah, end-to-end. But otherwise, Google jumped through some weird hoops to offer this.

Since the technology is interesting, and since it might well be of interest to our listeners whose corporations might find value here, I want to take us into the details. For that, ArsTechnica's Dan Goodin does a terrific job of setting this up, creating the appropriate context, and explaining what goes on. Ars' headline last week was: *"Gmail unveils end-to-end encrypted messages. Only thing is: It's not true E2EE."* and their tag line was *"Yes, encryption/decryption occurs on end-user devices, but there's a catch."* Dan writes:

> *When Google announced Tuesday that end-to-end encrypted messages were coming to Gmail for business users, some people balked, noting it wasn't true E2EE as the term is known in privacy and security circles. Others wondered precisely how it works under the hood. Here's a description of what the new service does and doesn't do, as well as some of the basic security that underpins it.*

I'll interrupt here for a moment to note that the way conventional end-to-end encryption operates is pretty straightforward. Each party has a public key pair, consisting of a public key and a private key. And the public keys are published in some way. So when Alice wishes to send an encrypted message to Bob, she first creates a high-entropy secret symmetric key which will be used to encrypt her message. So she uses that secret symmetric key to encrypt everything that she wishes to send to Bob. Next, Alice encrypts that secret key twice. First with her private key, then a second time with Bob's public key. She then packages the encrypted message up along with the result of the double key encryption and sends the package to Bob.

Upon receiving Alice's package, Bob first decrypts the double-encrypted key using his secret key, which only he knows. He then looks up Alice's publicly published public key and uses it to decrypt the result of the first decryption. Only if all four of these keys were correct will Bob now have recovered the properly decrypted secret symmetric key which he can use to decrypt the package that Alice prepared for him.

The elegant beauty of this simple system is that Alice wishes to send something that only Bob can decrypt and Bob wants to know that whatever he received was truly sent by Alice. Since both party's private keys must be used, and only each party knows their own, not only do we get strong encryption protection from anyone attempting to intercept the communication, but Alice knows that only Bob can decrypt what she encrypted and Bob knows that only Alice can have sent what he decrypted.

So what has Google done with Gmail? ... because it's certainly not that. Dan continues:

> *When Google uses the term E2EE in this context, it means that an email is encrypted inside Chrome, Firefox, or just about any other browser the sender chooses. As the message makes its way to its destination, it remains encrypted and can't be decrypted until it arrives at its final destination, when it's decrypted in the recipient's browser.*
>
> *The chief selling point of this new service is that it allows government agencies and the businesses that work with them to comply with a raft of security and privacy regulations and at the same time eliminates the massive headaches that have traditionally plagued anyone deploying such regulation-compliant email systems. Up to now, the most common means has been S/MIME, a standard so complex and painful that only the bravest and most well-resourced organizations tend to implement it.*
>
> *S/MIME requires each sender and receiver to have an X.509 certificate that's been issued by a certificate authority. Obtaining, distributing, and managing these certificates in a secure manner takes time, money, and coordination. That means that if Bob and Alice have never worked together before and an urgent or unexpected need arises for him to send Alice an encrypted message promptly, they're out of luck until an admin applies for a certificate and sees that it's installed on Alice's machine—so much for flexibility and agility.*
>
> *Google says that E2EE Gmail abstracts away this complexity. Instead, Bob drafts an email to Alice, clicks a button that turns on the feature, and hits send. Bob's browser encrypts the message, and sends it to Alice. The message decrypts only after it arrives in Alice's browser and she authenticates herself.*
>
> *To make this happen, Bob's organization deploys what Google says is a lightweight key server, known as a **KACL**, short for a **Key Access Control List**. This server, which can be hosted on premises or most cloud services, is where keys are generated and stored. When Bob sends an encrypted message, his browser connects to the key server and obtains an ephemeral*

> ***symmetric*** *encryption key. Bob’s browser encrypts the message and sends it to Alice, along with a reference key. Alice’s browser uses the reference key to download the symmetric key from the KACL and decrypts the message. The key is then deleted.*
>
> *To prevent Mallory or another adversary-in-the-middle from obtaining the key, Alice must first authenticate herself through Okta, Ping, or whatever other identity provider, or IDP, Bob’s organization uses. If this is the first time Alice has received a message from Bob’s organization, she will first have to prove to the IDP that she has control of her email address. If Alice plans to receive encrypted emails from Bob’s organization in the future, Alice sets up an account that can be used going forward.*
>
> *Bob’s organization can add an additional layer of protection by requiring Alice to already have an account on the IDP and authenticate herself through it.*
>
> *Julien Duplant, a Google Workspace product manager, told Ars: “The idea is that no matter what, at no time and in no way does Gmail ever have the real key. Never. And we never have the decrypted content. It’s only happening on that user’s device.”*

I’m going to interrupt here again to note that **in no way** is any web browser a safe place to decrypt super-secure, like national security level or extremely proprietary corporate, material. You still have JavaScript or WebAssembly running in a web browser which **is** as authentically secure as we’ve been able to make them, but they are still being updated to cure serious, often 0-day style security vulnerabilities. If you really need to send something securely, encrypt it offline away from any web browser then send it in the clear through any email system.

I’m not intending to take anything away from Google. The system they’ve created is an interesting hack, but a hack it is. And it also represents a security tradeoff for convenience since it’s running in the largest attack surface – today’s web browser – that any computer system has. Dan finishes his description, writing:

> *Now, as to whether this constitutes true E2EE, it likely doesn’t, at least under stricter definitions that are commonly used. To purists, E2EE means that only the sender and the recipient have the means necessary to encrypt and decrypt the message. That’s not the case here, since the people inside Bob’s organization who deployed and manage the* ***KACL*** *have true custody of the key.*
>
> *In other words, the actual encryption and decryption process occurs on the end-user devices, not on the organization’s server or anywhere else in between. That’s the part that Google says is E2EE. The keys, however, are managed by Bob’s organization. Admins with full access can snoop on the communications at any time.*
>
> *The mechanism making all of this possible is what Google calls* ***CSE****, short for* ***C****lient-****S****ide* ***E****ncryption. It provides a simple programming interface that streamlines the process. Until now, CSE worked only with S/MIME. What’s new here is a mechanism for securely sharing a symmetric key between Bob’s organization and Alice or anyone else Bob wants to email.*
>
> *The new feature is of potential value to organizations that must comply with onerous regulations mandating end-to-end encryption. It most definitely is not suitable for consumers or anyone who wants sole control over the messages they send. Privacy advocates, take note.*

If you may have seen some news about Gmail’s new E2EE, now you have some context.

CVSS 10.0 in Apache Parquet
Apache recently received the much dreaded full CVSS 10.0 with a widely used module known as Apache Parquet (spelled Parquet). Apache Parquet is an open-source, columnar storage format designed for more efficient data processing. Unlike row-based formats (like CSV), Parquet stores data by columns, which makes it faster and more space-efficient for analytical workloads. It is widely adopted across the data engineering and analytics ecosystem, including big data platforms like Hadoop, AWS, Amazon, Google, Azure cloud services, data lakes, and ETL tools. Some large companies that use Parquet include Netflix, Uber, Airbnb, and LinkedIn.

And now a new, low-complexity, remote code execution vulnerability has been identified in all current versions of the system. Unfortunately, the problem was disclosed on April 1st, but since this is no joke, and it would be horrible for those affected, I hope no one dismissed it out of hand.

So this maximum severity remote code execution (RCE) vulnerability impacts all versions of Apache Parquet up to and including 1.15.0. The problem stems from the deserialization of untrusted data (also known as interpretation) that could allow attackers with specially crafted Parquet files to gain control of target systems, exfiltrate or modify data, disrupt services, or introduce dangerous payloads such as ransomware. The vulnerability is tracked as CVE-2025-30065 and, as I said, carries a CVSS v4 score of 10.0. The flaw was fixed with the release of Apache version 1.15.1.

It's some solace that in order to exploit this flaw, threat actors must convince someone to import a specially crafted Parquet file. But we all know that social engineering attacks remain some of the hardest to defeat. And it might be that there are other entry vectors.

A bulletin by Endor Labs highlights the risk of CVE-2025-30065 exploitation more clearly, warning that the flaw can impact any data pipelines and analytics systems that import Parquet files, with the risk being significant for files sourced from external points.

Endor Labs believes the problem was introduced in Parquet version 1.8.0, though older releases may also be impacted. The firm suggests coordinated checks with developers and vendors to determine what Praquet versions are used in production software stacks.

Endor Labs wrote: "If an attacker tricks a vulnerable system into reading a specially crafted Parquet file, they could gain remote code execution (RCE) on that system." However, they do avoid over-inflating the risk by including the note, "Despite the frightening potential, it's important to note that the vulnerability can only be exploited if a malicious Parquet file is imported."

That being said, if upgrading to Apache Parquet 1.15.1 immediately is impossible, it is suggested to avoid untrusted Parquet files or carefully validate their safety before processing them. Also, monitoring and logging on systems that handle Parquet processing should be increased.

Although no active exploitation has been discovered yet, the risk is high due to the flaw's severity and the widespread use of Parquet files in big data applications. Consequently, administrators of impacted systems are recommended to upgrade to Parquet version 1.15.1 as soon as possible.

Endor Labs noted that while there are currently no known instances of exploitation and no published proofs of concept, now that the bad guys know of this juicy new vulnerability the race will be on to find the flaw and see about leveraging it for attacks.

Listener Feedback

@TechnoAgorist

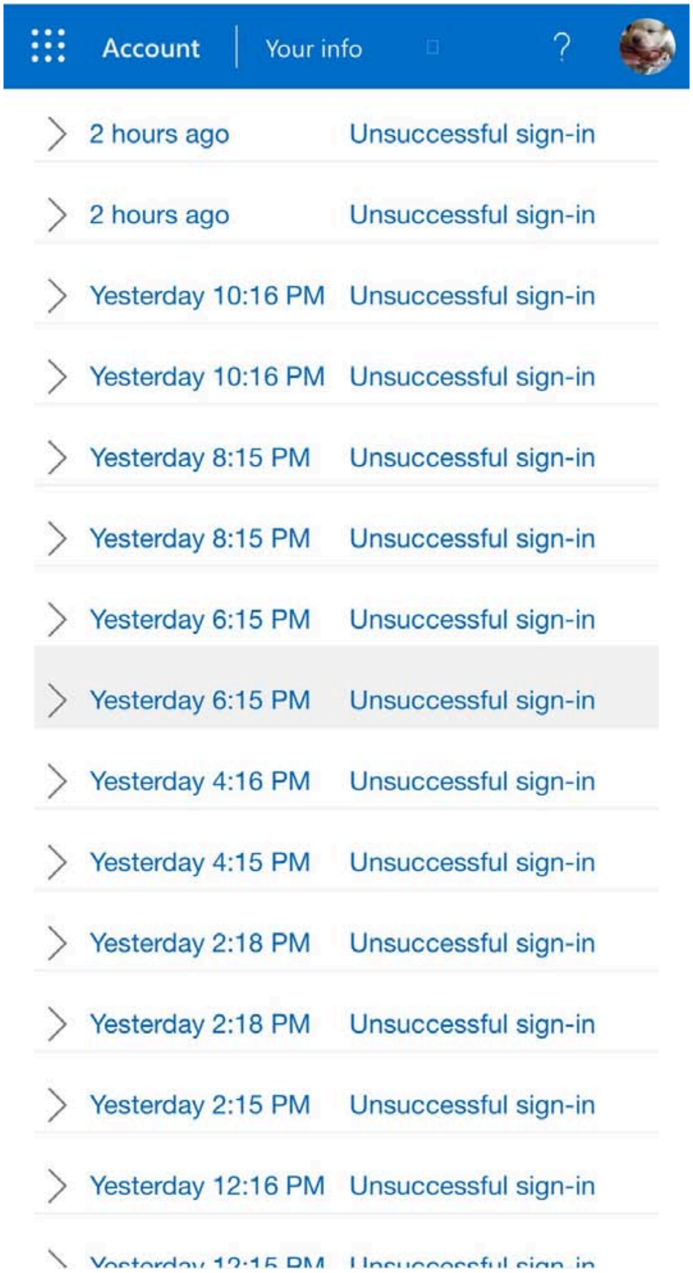
Regarding Neal Asher's novels, they may not be on Kindle Unlimited but I found them at my local library. That is how I have been reading them. Thanks for the recommendation.

I appreciated being able to share a reminder about printed books. I'm on Book #4 of the first 5- book Agent Cormac series and I'm having a great time. The books are long and involved. The style Neal used for the first three was to create several parallel plot lines that initially bore no obvious relationship to each other. But as the story progressed they eventually merged into an intriguing conclusion. Although this is a common organization for authors, Neal's use of this somehow feels more explicit. Perhaps it's that I was initially trying too hard to work out what one thread had to do with the other. Now, having seen that's what he does, I'm content to let it just happen organically.

@Eric Seidel

Steve: Hey I just listened to part of your podcast and it was funny that you mentioned something that happened exactly to me as well. In the past couple of days, I had Microsoft 2FA reset requests show up in my email and then happened to look in my sign in activity and it is sign in request every minute to my account it's just insane.

Make sure you have your 2FA turned on. Holy smokes!



2 hours ago	Unsuccessful sign-in
2 hours ago	Unsuccessful sign-in
Yesterday 10:16 PM	Unsuccessful sign-in
Yesterday 10:16 PM	Unsuccessful sign-in
Yesterday 8:15 PM	Unsuccessful sign-in
Yesterday 8:15 PM	Unsuccessful sign-in
Yesterday 6:15 PM	Unsuccessful sign-in
Yesterday 6:15 PM	Unsuccessful sign-in
Yesterday 4:16 PM	Unsuccessful sign-in
Yesterday 4:15 PM	Unsuccessful sign-in
Yesterday 2:18 PM	Unsuccessful sign-in
Yesterday 2:18 PM	Unsuccessful sign-in
Yesterday 2:15 PM	Unsuccessful sign-in
Yesterday 12:16 PM	Unsuccessful sign-in
Yesterday 12:15 PM	Unsuccessful sign-in

Matthew West

Hi, Love the show. I bought a used Fitbit with a cracked screen. I forgot that I would need the PIN shown on the screen to pair. I am trying to pair by constantly changing one time code in the hopes it eventually works. This made me wonder what the best strategy is and how many attempts would be needed to reach a 50% chance. Sorry if this was already answered. I should look through the transcripts. Thank you.

We previously addressed this question a few months back, when we took our deep dive into the precise operation of hash-based one time passwords. The podcast was #1009 and we received an unusual amount of positive feedback from our listeners who enjoyed thinking about the various aspects of a 6-digit code that was changing randomly every 30 seconds.

The answer to the first part of your question *"what's the best strategy"* is that since the proper PIN code at any given instant is likely completely random, there can be no *"best strategy"* since no guess can, by definition, be any better than any other. So if "patience" could be considered a strategy then "patience" would be best because a great deal of patience is what would be needed. So, exactly how much patience? The second part of your question asked *"how many attempts would be needed to reach a 50% chance?"* And that's something that's knowable.

At the bottom of page 21 of episode #1009's show notes, I wrote:

> *The probability of things happening is something that often trips people up. If the probability of something random happening is one in a million, we might tend to assume that giving that one-in-a-million thing one million opportunities to occur – or in our case one million key guesses – we would* ***probably*** *obtain a collision of 6-digit values. And that's true, but it's not guaranteed. Probability theory tells us that even given one million guesses of a one-in-a-million event, there's a 36.79% chance of never hitting upon the value we're seeking. But that means that given one million guesses there's a 63.21% chance – so well better than 50/50 – of hitting upon the number we're looking for. But 63.21% means that it's not a certainty.*
>
> *For random events it's all about probabilities and 693,147 guesses – so nearly 700,000 – would be required to hit the 50/50 point – for an even chance of any one-in-a-million guess being correct.*

So that's the answer to your question, Matthew. Which is why "patience" will be the best strategy. You'll need to make 693,147 individual PIN number guesses for there to be an even 50/50 chance of pairing your Fitbit with the broken screen to your smartphone.

Jason

> *Hi Steve & Leo, Long time listener and happy ClubTWiT member. As we all move to delete our 23andme data, I have a maybe-amusing story. When I signed up for 23andme years ago, I thought I would attempt to get privacy by obscurity. I created my 23andme account with a fake name, with a new gmail for that fake name. My thought was if they were ever hacked, as they were, or sold their data, as they are, at least my DNA wouldn't be tagged with my name. So I also made up a fake birthday, in keeping with the obscurity strategy.*
>
> *Cut to this week when I went to delete my data and found that 'birthday" is used as a form of authentication. I have no idea what date I gave them and I never thought to record it. I tried permutations of my own birthday until I ran out of guesses and locked myself out. Emails to their support revealed that the only way to prove my identity was to provide government issued ID. I'm not likely to give my ID to someone actively selling all of their assets to the highest bidder anyway, but I certainly can't when no such ID exists. Oh well, guess I'll have to continue to rely on obscurity. Thanks for all you do, "Jason"*

I loved that "Jason" also put his own name in quotes... suggesting that he's quite deeply committed to remaining anonymous and obscure – as indeed he is. And given that no one knows whose DNA his is anyway, let alone who he is, I'd say there never was any need to delete it in the first place. But I understand for the sake of "why not delete it" it made sense to try.

(Anonymous) *wanted to share some thoughts about leaving Windows…*

> *Hi Steve, Please keep my name, company and project private because it would be easy to reverse engineer who my company is. I've been listening for years, thank you for all you do! I'm a security researcher and developer at [really big company X]. I mostly maintain a popular open source tool [name redacted].*
>
> *With respect to moving away from Windows to an open source solution: Much of my company's software's (firmware) build chain is built upon Windows. Microsoft is in the process of re- licensing all of our Server Win OS and MS SQL agreements and as a result our cost would be going from a per compute device license to a per core license. As such, the cost would be going from thousands of dollars to millions of dollars. In response, we are simply moving as much of our infrastructure as we can to an open source variant.*
>
> *It seems crazy to me that M$ is so arrogant that they think there's no alternative to them, or at least that the cost would be too much for us to absorb. About that, they have miscalculated. Yes it will cost us to move, but it'll be* ***so*** *nice once we've done so. Now we just need to move all of our clients from Windows to Linux and I'll be a happy camper. Thanks again for all you do! / Anon.*

This person was just one of many of our listeners who wrote to me in response to last week's EU OS podcast. I heard similar stories over and over and over. Microsoft apparently believes that they will be maximizing their bottom line profit by squeezing more money out of fewer customers because the theme that I heard playing out over and over was that people were finally and at long last throwing in the towel, giving up, and biting the bullet to move to free and open source solutions. Those solutions have been steadily maturing through the years and are finally solid enough to be depended upon. And the message was: moreso than Microsoft, whose policies appear to be predatory.

TJ Asher

> *Steve, Heard Leo mention Jackpot Junction in that list of companies on the ransomware site. That's a casino here in Minnesota. So I went to their website and they have a big notice:*
>
> > Slot machines and kiosks are currently unavailable. Bingo is canceled until further notice. The special Bingo session is postponed until a later date. Continuity is postponed until further notice. Promotional drawings are postponed until further notice.
> >
> > Dacotah Dining is closed until further notice. Full Deck is open for breakfast from 7 am - 11 am with regular menu from 11 am - close.
> >
> > Table Games and Circle Bar will remain open.
> >
> > Thank you for your patience and understanding. We will provide updates as they are available.
>
> *Definitely looks like they got hacked. Keep up the awesome work! Regards, TJ Asher*

Thanks, TJ. It's fun to have that confirmation of the data being continually posted on the https://www.ransomlook.io site. And remember, you can find it at grc.sc/019.

Henrik Johnson

Hello, I just thought I'd clarify something you and Leo said in episode 1019 about CloudFlare hosting 20% of the web. The 20% figure most likely refers to sites behind CloudFlare WAF not actual hosting (Especially since they referred to their free plan which does not include hosting). That said, when behind a WAF CloudFlare does terminate TLS, which means that they are an intentional man in the middle that can see request information including login credentials.
/Henrik

Thank you, Henrik for clarifying that. I should have been more clear. A better way to say it would be that Cloudflare is "fronting" for 20% of the Internet's website properties.

Harry Pilgrim

Steve, You and Leo continue to say that you use "certificates" to login to SSH servers. This is not completely accurate. SSH can be configured to use public/private keys for authentication, but these are not "certificates."

A certificate is composed of uniquely identifying information such as FQDN, name, company, state, country etc AND your public key. These details are verified by a Certificate Authority and attested to be genuine by the CAs digital signature, and the fact that our systems trust that CAs assertion.

Part of the certificate is the public key that the user provides in the Certificate Signing Request, but the certificate includes much more than just the public key.

Some SSH servers CAN be configured to use X509 certificates, but this is more complicated and requires deep knowledge of PKI, such as implementing certificate revocation checking on the SSH server. Out of the box, none of the Linux SSHd implement X509 authentication by default, and only a few SSH clients I've found, such as SecureCRT and a particular version of PuTty, support sending X509 certificates through the ssh protocol.

Harry Pilgrim

Harry, thank you for correcting us. And you are, of course, 100% correct. We should have been saying that we authenticate our SSH sessions using public key crypto with large and long keys, not certificates.

This gives me the opportunity to mention my absolute favorite SSH client and server solution for Windows-centric users, which is Bitvise: https://bitvise.com/. They're not a new discovery of mine because I would never recommend something like an SSH client and server without first obtaining sufficient experience for any such recommendation. But I've now been using their solutions since 2018, so I've gained seven years of experience with their software and their company... and I could not recommend them more highly.

If all you need is an incredibly good SSH client for Windows, you can use theirs free of change. The Bitvise client is free. If you also want a terrific SSH server for Windows, you can take theirs out for a 30-day spin for free after which a one-year license is $100, but only access to upgrades expires. The server will run forever. And having been with them for 7 years, I can attest that they are not constantly fixing their mistakes. Their SSH server is bristling with useful features and their SSH client is a joy to use. I could not be more pleased with them and I cannot imagine ever having a need to switch. So, just for the record: BITVISE is my SSH solution for Windows.

And back to Harry’s correction about Leo and my use of public and private key crypto for authentication, Bitvise handles all of that beautifully.

David Spicer

> *Steve, I was listening to podcast episode #1019 and as you talked about Troy Hunt getting phished, I couldn't help but wonder how one could help prevent this type of quick acting attack. I know Passkeys would solve a lot of this in the first place, but I often see cloud services that support Passkeys also allow for username and password as a backup. I personally find it difficult to see how sites that support both options are any safer, but that's another issue.*
>
> *My online banking site requires an OTP code just to login. Once in, I can view all of my account information like normal. However, if I want to perform any money transfers, I am prompted for a new OTP code before I can do so. That made me think that this method might be useful with other online services that only support OTP MFA such as MailChimp. Even after you have signed in, if you wanted to perform a security relevant action, such as exporting data, changing authentication methods, or viewing API keys, that would require a new OTP code from your authenticator. This would help prevent attackers who phish a login from you from being able to make changes or steal sensitive information without having to phish for a second OTP code from you. Well, that was just my thought anyways.*
>
> *I'm glad I found your podcast nearly a decade ago, I love listening to you and Leo every week. Every episode is a good one and your tools like SpinRite, ValiDrive, and DNS Benchmark are amazingly useful! Really looking forward to buying the Pro version of the DNS Benchmark when that comes out for my lab environment. Have a great week! Thanks, David*

I agree with David. Requiring the re-use of a one-time-password before proceeding with any extra sensitive action makes a ton of sense. It’s exactly analogous to pretty much any site asking us to re-supply our current password as part of the process of changing that password. Why? We’re obviously already logged-in, in order to even be presented with that opportunity. The site already knows who we are enough to allow us to be roaming around. So why ask us to reassert our current password before we’re able to change it? Obviously, because changing our password is seen as a particularly sensitive action. But to David’s point, it’s interesting that this re-use of one-time-passwords does not seem to have filtered down into the operation of most sites beyond login authentication. His bank and others being a common exception.

My own presumption is that the reason for this is that most sites are using some canned OAuth logon authentication solution and haven’t bothered to build-in one-time-password re-verification. Perhaps in time that will change since re-prompting for one-time-passwords makes so much sense.

John Rostern

> *Steve, I've been a long time Security Now listener and have always appreciated your insightful commentary, and analysis (mixed with some humor) on all things related to cyber security. I was a bit taken aback therefore by your somewhat dismissive comments regarding the Security Technical Implementation Guides (STIGs) in #1018. The STIGs (https://public.cyber.mil/stigs/) represent an authoritative resource for secure systems deployment. The voluminous STIG documentation and tools are all provided free of charge including the Security Content Automation Protocol (SCAP) benchmarks. Misconfiguration has*

> *been and remains an primary threat vector and following guidance such as that provided by the STIGs or the CIS Benchmarks in the deployment process is a critical preventive control.*
>
> *Your show is a valuable resource for security practitioners that helps elevate the state of the practice across the community. It would be a disservice to minimize the potential value of a resource such as the DISA STIGs. Kind Regards, John Rostern*

Thank you, John – I stand before you willingly chastened. I did not intend to be dismissive of the STIGs because I am not at all familiar with them at all. But I'm always wary of bureaucracy and, by extension, the trappings of bureaucracy. This is why, for example, I've been so pleasantly surprised by the value and effectiveness of CISA. "Value and effectiveness" is never what I expect from governmental agencies – especially cyber-agencies. So, thank you for correcting me on the matter of the value of the STIGs. For anyone who's interested in these Security Technical Implementation Guides, I have the link to them, which John provided, in the show notes.

Michael Swanson
It appears that many of our listeners have encountered these STIGs...

> *Hi Steve,*
>
> *In a recent episode Dan Linder brought Security Technical Implementation Guides (STIGs) to your attention. I thought a little more info might be useful to your listeners as STIGs are very useful in hardening systems against threat actors.*
>
> *These STIGs are created and maintained by the US Department of Defense in cooperation with the manufacturers and developers of various hardware and software. They are reviewed and updated continuously with a quarterly publishing cycle. STIGs exist for a wide variety of hardware devices (most notably firewalls and network switches), operating systems (Windows, MacOS, various Linux distros, VMware, iOS, Android, etc.), web browsers (Chrome, Firefox, etc.), common applications (MS Office, Adobe, etc.), even Active Directory (one of the most important if you want to keep attackers from moving laterally in your network).*
>
> *As Dan mentioned, some of the settings are policy and procedure (user accounts are deleted from the system when an employee leaves the organization), while others are technical (two factor authentication is required to access the system). Bottom line, these checklists of settings work. Searching for 'DISA STIG' will take your listeners to the library.*
>
> *Best regards,*
> *Mike Swanson*

Thank you, Michael. This makes sense. I went over to https://stigviewer.com/stigs and took a look around. There is a lot of interesting security content organized by the name of the hardware or software that's the topic of each of the many individual *Security Technical Implementation Guides*. You can go to https://stigviewer.com/ to see the most recent entries and then choose "STIGS" from the upper-right top-of-screen menu to see a huge alphabetically sorted list of very useful security-hardening checklists.

Thanks guys!

Multi-Perspective Issuance Corroboration

Today's main topic was an outgrowth of an interesting change that the famous CA/Browser forum just ratified. The CA/Browser forum consists of those people who determine what criteria are needed for web browser certificate issuance, how long various issued certificates will be permitted to live, how browsers will deal with such certificates and everything else that's relevant surrounding the increasingly crucial need for the clients on the Internet – whether they be people or automated systems – to be assured that the servers they're communicating with at the other end somewhere else, anywhere else, in the world are really the entity they claim to be.

A couple of weeks ago the CA/Browser forum agreed to significantly up the ante – for all Certificate Authorities everywhere – on one crucial aspect of the mechanism that is relied upon for verifying the ownership and control of the domains for which certificates are being issued. I first learned of this from Google's announcement of this news, they wrote:

> *The Chrome Root Program led a work team of ecosystem participants, which culminated in a CA/Browser Forum Ballot to require adoption of* ***MPIC*** *via Ballot SC-067. The ballot received unanimous support from organizations who participated in voting. Beginning March 15, 2025, CAs issuing publicly-trusted certificates must now rely on MPIC as part of their certificate issuance process. Some of these CAs are relying on the Open MPIC Project to ensure their implementations are robust and consistent with ecosystem expectations.*

So something recently happened in the world of web server certificate issuance. This whole area is a fascinating subject which this podcast has spent time examining through the years. So what exactly is MPIC? Here's how Google explains it:

> *Before issuing a certificate to a website, a Certification Authority (CA) must verify the requestor legitimately controls the domain whose name will be represented in the certificate. This process is referred to as "domain control validation" and there are several well-defined methods that can be used. For example, a CA can specify a random value to be placed on a website, and then perform a check to verify the value's presence has been published by the certificate requestor.*
>
> *Despite the existing domain control validation requirements defined by the CA/Browser Forum, peer-reviewed research authored by the Center for Information Technology Policy (CITP) of Princeton University and others highlighted the risk of Border Gateway Protocol (BGP) attacks and prefix-hijacking resulting in fraudulently issued certificates. This risk was not merely theoretical, as it was demonstrated that attackers successfully exploited this vulnerability on numerous occasions, with just one of these attacks resulting in approximately $2 million dollars of direct losses.*
>
> *Multi-Perspective Issuance Corroboration (referred to as "MPIC") enhances existing domain control validation methods by reducing the likelihood that routing attacks can result in fraudulently issued certificates. Rather than performing domain control validation and authorization from a single geographic or routing vantage point, which an adversary could influence as demonstrated by security researchers, MPIC implementations perform the same validation from multiple geographic locations and/or Internet Service Providers. This has been observed as an effective countermeasure against ethically conducted, real-world BGP hijacks.*

Let’s clarify this. In order to really understand the problem, we need to first revisit the operation of the Internet at its most fundamental level. It’s been a long time since we’ve done that, so let’s first do a quick bit of review about exactly how the Internet works.

As we discussed way back in the dawn of this podcast, the brilliant way the Internet works and the thing that has ultimately been wholly responsible for the Internet’s robustness, is that it has never tried to be perfect. Its original brilliant design replied upon a “best effort” packet routing system. In this system, data to be sent from point A to point B was first “packetized” by breaking anything larger than a packet, of around 1500 bytes, into multiple individual packets. Each individual packet indicates where it is from and where it hopes to go. The packets are then dropped one by one onto the Internet.

The Internet itself, as we’ve come to know it, consists of a massive network of so-called “big iron” Internet routers each of which is connected to a bunch of its neighboring routers. As each of these routers has multiple high bandwidth interfaces each of which connects to other similarly well-connected Internet routers. So the Internet itself is actually nothing more than a huge global quilt of large industrial-strength routers, each of which is interconnected to its nearest neighbors in a huge, largely ad hoc, array. The Internet’s users are individually connected to one of these big local Internet routers by their ISP, which then drops their packets onto the big iron router that’s run by the ISP.

Upon arriving at the first Internet router, that router obtains the packet’s requested destination, then looks up the destination in its own routing table to determine which of the many other big iron Internet routers it should send that packet to in order to move that packet closer to its requested destination. So the packet is then forwarded to that next router which moves it closer to its intended recipient.

These individual routers have receiving buffers on their interfaces which allow incoming packets to queue up while they’re waiting to be forwarded. But it might happen that too many packets arrive from too many different interfaces, all requesting to be forwarded out through the same destination interface – and that might not be physically possible. In that case, the router’s incoming packet buffers would overflow with nowhere to temporarily store newly arriving packets, and those packets would be dropped and lost forever.

At first this might seem like a very bad thing. Like a critical flaw in the design of this system. But it turns out that this reflects the original brilliance of the Internet’s designers. They said “Okay, no, that’s not good. So let’s make it survivable. Let’s design the protocols that place these individual potentially lossy packets onto the Internet in such a way that packet loss is okay.

So, for example, in the case of the UDP protocol being used for DNS lookup. If an answer to a query for a domain’s IP address is not received within a reasonable amount of time, the query will be retried and reissued and other DNS servers will also be asked. And this will continue until a reply is received.

So, crazy as it might at first seem, every Internet protocol that generates and receives individual Internet packets assumes that its packets might not arrive at the other end and arranges for that possibility. This brilliant design decision takes the pressure off the Internet’s packet delivery system which is simply a massive ad hoc network of loosely interconnected routers. This allows them to do the best job they can of receiving packets on their various interfaces and sending them along their way toward their destination by routing them out of other interfaces. And if incoming packet buffers overflow, that’s not the router’s problem. The protocol which originally generated the packets will deal with that.

Okay. So what does this have to do with BGP?

This massive network of interconnected routers need some means of knowing which IP address ranges should be sent out of which of their many interfaces. To answer this question, each router contains a routing table to specify which addresses can eventually be reached through which interface. How are these big routing tables determined and maintained? That's where the Internet's BGP, the Border Gateway Protocol, comes in. BGP is used by the Internet's big-iron routers to coordinate, synchronize and update their understanding of which packets should be sent where.

An ISP's big-iron Internet router uses BGP to "advertise" the various blocks of IP addresses it has been assigned by the Internet's governing bodies and which its customers are using. BGP sends this to all of the routers that connect to the ISP's router so that they know to forward any packets they receive on any of their other interfaces to the interface connected to the ISP's router. After setting up their own routing tables appropriately, each of those routers, in turn, use BGP to forward their own routing tables to the neighbors they connect to ... and so on and so on ... until eventually every big-iron router anywhere on the Internet has received the information about where to send any packets that are destined for that ISP's big-iron Internet router.

And believe it or not, this entire system works, and works with astonishing reliability. When it fails, failures are generally local and are quickly fixable. The system is not perfect. Through the years we've covered the news of innocent mistakes made with the Internet's big routers which for a few very hectic minutes might attempt to route all of the entire Internet's traffic through a bungalow in Myanmar. But perfection is understood to be impossible, so a system that's self healing and resilient in the face of mistakes is what we have today. And also through the years, the original vulnerabilities in these systems have been recognized, shored up, and improved.

This finally brings us back to the rules change that the CA/Browser recently enacted. In order for me to obtain a TLS certificate from DigiCert for the grc.com domain, I need to demonstrate that I'm in control of the grc.com domain. So DigiCert gives me a simple file with a random gibberish name for me to place in the root directory of my web server at grc.com. Once I've done so, I let DigiCert's automation know and it attempts to obtain that file by that name and with the proper contents from the root of grc.com. If that can be done, that proves to DigiCert that I'm able to affect the content of the website at grc.com (which no one else can do) and thus I'm allowed to obtain an identity certificate which covers that domain.

But here's the problem: When DigiCert's automation reaches out to my webserver at grc.com, it's just sending packets to its ISP in Utah, which then drops them onto its big-iron Internet router for them to then be sent from Utah to my ISP in California and then to GRC's web server. In other words, DigiCert in Utah connects to my webserver in California which has the IP address of grc.com and verifies the contents of a specific file which they created for that purpose.

The implicit and crucial assumption is that the packets DigiCert caused to be dropped onto the Internet in Utah were actually routed to and received by the webserver at GRC.COM in California. Everything about the legitimacy of the certificate GRC has requested depends and relies upon the truth that DigiCert obtained that file from my webserver and not from someone else's.

A so-called BGP Prefix Attack involves someone arranging to insert the network prefix for a small network into a big iron Internet router which would then cause it to misroute any packets bound for any IP addresses within that small prefix network. In other words, the traffic for a specific network would be effectively hijacked.
Following further with our example, if this were done to a router near DigiCert through which the

packets bound for GRC was traversing, those packets would be sent not to GRC but presumably to an attacker. In doing this, the attacker’s server (not mine) would be hosting the domain control validation file and they would be proving that they, not I, control the grc.com domain. And DigiCert would then, having done their due diligence, issue them a web server TLS identity certificate for the GRC.COM domain.

And here’s the crucial point:

The only way and reason this BGP router prefix-hijack attack works is that a router close to DigiCert, through which an attacker was certain DigiCert’s packet traffic destined for grc.com would be flowing, could be compromised. While this compromise was in place, and my webserver at GRC.COM was effectively unreachable by DigiCert, it would still be reachable by everyone and anyone else located anywhere else through other non-compromised routers.

And this brings us to the need for MPIC “multi-perspective issuance corroboration”. With the researchers at Princeton University’s Center for Information Technology Policy (CITP) having demonstrated the real world feasibility of these BGP prefix-hijack attacks, all Certificate Authorities going forward will need to perform domain control validation from multiple geographically diverse locations.

Immediately, as of March 15th last month, validation must be made from at least two remote network perspectives. Ca’s have a year to bring that number up to 3 and from at least two distinct Regional Internet Registry (RIR) regions. By June 15th of next year, 2026, that number grows to 4, also from at least 2 Regional Internet Registry regions, and by the end of next year, December 2026, at least five remote network perspectives must be used.

Five! Wow. So it’s clear that once again, these guys are not taking any chances. It would be so supremely difficult to somehow arrange to simultaneously intercept traffic originating from as many as five locations that it’s safe to say that this takes this mode of validation attack off the table.