Multi-Perspective Issuance Corroboration

Description: Canon printer driver vulnerabilities enable Windows kernel exploitation. Astonishing cybersecurity 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?

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-1020.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-1020-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about, including a 10.0 CVSS score for a problem in Apache Parquet. French schoolchildren are not gullible, it turns out. The French government tried to trick them and failed. And then we'll find out what multi-perspective issuance corroboration is and why you might need it. That and a whole lot more coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1020, recorded April 8th, 2025: Multi-Perspective Issuance Corroboration. It's time for Security Now!, the show where we talk about your safety, your privacy, your security, and a bunch of other stuff that geeks are interested in with this guy right here, I think you officially are the King of Geeks, Steve Gibson.

Steve Gibson: I would wear that badge proudly, Leo.

Leo: Wouldn't you? Yeah.

Steve: Yes, I would.

Leo: You have earned it over the years, now 70 of them. Congratulations. That's amazing.

Steve: We had a listener who had a T-shirt made and sent me a photo, "Just Say No to Port 80."

Leo: I love it. I love it.

Steve: From last week's podcast. Which reminded me, I had some made a while ago that just said "Born to Code."

Leo: Yes.

Steve: Because I put on - I make a fresh cup of coffee, put on some quiet music, sit down in front of my computer, and, like...

Leo: It's the best, isn't it.

Steve: Some problems to solve. It is my happy place.

Leo: Me, too.

Steve: There's, like, nothing like it.

Leo: Yeah.

Steve: That is just - that's it.

Leo: I get sad when I hear about Vibe coding and AI replacing engineers because I think it is, independent of whether it's a useful economic exercise, a wonderful, fun thing to do.

Steve: A buddy of mine sent a link yesterday to a blog post. There's a guy named, I think it's Ken Schiff, but that sounds like it's, like his last name, I've shortened it. Anyway, he's been - he, like, pops the lids on Intel chips, Intel processors.

Leo: Oh, wow. Oh my god.

Steve: And then takes photomicrographs of the chip and then reverse engineers the circuitry. And this particular blog posting was about the times three multiplier hardware in the Pentium that was like, they had like a dedicated strip of silicon that was for multiplying by three. That's all it did. And then it was like, okay, why is there a hardware times three multiplier? And actually it's not difficult to multiply by three in binary, right, because you just shift over by one, and sum. But it turns out that they don't do binary multiplication. They do base 8 multiplication in the hardware. It's like, what?

Leo: What?

Steve: Anyway, so this guy has just gone so into this. And so I read the blog posting, and I thought, you know, this kind of thing, the designers of that never were understood. They didn't get credit for it. I'm sure they were working within a small community of just incredible silicon design wizards. And hopefully that was all they needed. But there's just - there's like this incredible wizardry in stuff back then. And it does feel like those days are leaving, sort of like turning coding over to AI. Unfortunately, coding makes so much sense for AI because it is so rigorously logical and so complex. And you should be able to say to an AI, does this do what I want? And it would just say, no, not even close.

Leo: Well, you know, it's a computer talking to a computer. Of course it kind of makes sense that a computer would do that well.

Steve: Yeah.

Leo: But at the same time it's, you know, people in grade school, I know everybody said, well, what am I learning, you know, algebra for; right? I'm never going to use algebra. It's not about the algebra, it's about the pleasure of it and kind of the formal reasoning, which is a great thing to learn.

Steve: Right.

Leo: I think everybody - I think coding should remain in the curriculum, even if it's not something you end up using.

Steve: Well, and teaching math is much the same way. It's just it's good for your brain.

Leo: Yeah, yeah.

Steve: To think, you know, in abstractions some of the time. Which is why we have this podcast, Leo. This is Security Now! Episode 1020.

Leo: Wow.

Steve: For Patch Tuesday. We actually have a picture that is apropos. I think you're going to enjoy it when we get to it. Today's title, I wasn't sure it was going to fit. Actually it strained the margins of the show notes.

Leo: It's a little long, and also a little obscure, I might add.

Steve: It's very obscure, yes. And it wasn't until my own description of the backstory behind this grew that I thought, well, this is our topic for the week: Multi-Perspective Issuance Corroboration, MPIC.

Leo: Okay.

Steve: And why, as of the middle of last month, the CA/Browser Forum, you know, the people who manage the certificates, the issuance and the consumption for web browsers, why they unanimously voted to require themselves to do this: Multi-Perspective Issuance Corroboration. So this is a big change that just happened in the requirements for issuing web browser certificates. Which we're going to get to after we look at Canon printer driver vulnerabilities enabling Windows kernel exploitation, and the astonishing cybersecurity awareness which has been shown by a household appliance manufacturer.

A listener pointed me to this company, I think they're Australian, or maybe they're - I don't remember where they are. Yeah, New Zealand, maybe? Anyway, unbelievable that they have a page because they're into connection and connected appliances, they understand what their obligation is if they're going to do it, like none other. Also, France tried to hook 2.5 million schoolchildren in a phishing test. We're going to look at the results of that. WordPress three years ago added an abuse-prone feature. Any guess what happened? Oh. And Oracle, is there something you would like to tell us that you have not so far?

Leo: Got to keep it a secret, just between us.

Steve: Yeah, some problems over there. And they're like, what? No. Nothing to see here. Just, you know, what's that big lump under the carpet? Don't worry about that.

Utah's governor just signed the App Store Accountability Act into law. We've talked about the legislation passing through their lower bodies. It's now law in Utah. Now what? Also it turns out that AI bots hungry for new data are inadvertently DDoSing FOSS projects.

Leo: Yeah. This is a problem, yeah.

Steve: Wow. Also, no Microsoft Account? No Microsoft Windows 11. A change has been made to the Dev Channel, coming soon to your next Windows 11 installation. Also, Gmail claims it now offers end-to-end encryption. Well, it kinda sorta does, somewhat. It is the definition of a hack, and we'll talk about it. Also, a dreaded CVSS 10.0 was discovered in something called Apache Parquet. Not good.

Leo: Butter. Oh, sorry. Sorry. I've been programmed.

Steve: But 10.0, everybody, so that's as bad as it gets. We've got a bunch of listener feedback. Believe it or not, I had time for that, too. And then we're going to look at what is Multi-Perspective Issuance Corroboration, and why must all Certificate Authorities now do it? And of course we've got a great Picture of the Week. So I think maybe, Leo, this podcast will finally be a good one.

Leo: Finally, after 1019 episodes.

Steve: I think we've got the hang of it now. There are people who...

Leo: You know, Sunday you should stop by and say hello. Our 20th Anniversary TWiT is this Sunday, after 20 years. Patrick Norton's going to come by. Sam Abuelsamid will be on, Allyn Malventano, and we're getting videos from all of our viewers. I've been asking everybody if you want to share your memory of the first time you saw TWiT or the first time you saw me and Steve, maybe back in the Screen Savers days. Share a video with us. We've got a lot of them. It's going to be a lot of fun. That's on Sunday. Can you believe it? Long time we've been doing this, Steve.

Steve: Well, and I asked Benito, I said, I thought that the number of Sundays TWiT was 1027.

Leo: It is, I think.

Steve: And today is 1020 for us. So Security Now!...

Leo: We're just a little ahead.

Steve: ...only started seven weeks later.

Leo: You're right. Well, maybe because you never stop. You know, for the first 15 years you wouldn't even take the Christmas holiday off. So, and maybe we missed...

Steve: It was that tattoo. That did it. I thought, okay, I'm quitting Christmas from now on.

Leo: There might be a few day - but, yeah, roughly seven weeks later. It was very quickly after it, yes, yes. And you're coming up on your 20th; right? When is that going to be? Do we know?

Steve: I don't know.

Leo: August, I think, yeah.

Steve: Yeah.

Leo: Twenty years. I don't feel that old. I really don't. It doesn't, you know, we started doing this in our late 40s.

Steve: What's cool is that we have really been on the podcast through huge changes in the industry.

Leo: Yeah.

Steve: I mean, like, you know, viruses moving from one person's thumb drive to the next, or computer to computer. I mean, that was a thing. And, you know...

Leo: There's a great movie just came out called "Black Bag." I don't want to spoil it. It's Michael Fassbender and Cate Blanchett, and you should watch it. Have you seen it?

Steve: No.

Leo: But the only reason I mention this is there's a moment when they're talking about this exploit that is a deadly exploit, and they said it's based on Stuxnet, and we've designed it for air-gapped computers. And I was thinking, man, they must listen to Security Now!. It was a really - it was technically a really great moment in that movie. It's a fun spy movie. But, you know what, that's one of the things, I think, maybe you could take a little bit of credit for. Hollywood is a little more savvy in the content, the computer content that you see onscreen.

Steve: Been very impressed with what they're doing now. I just think that it's percolated down into the culture.

Leo: The people who are writing this now are part of the...

Steve: Or they actually know we need to get a tech guy to, you know, help us with the script.

Leo: Yes.

Steve: And so there's some script polishing going on. There was a series, and I meant to mention it, except it wasn't that good, but it was about prime factorization.

Leo: Yes. I was going to ask you - I haven't watched it. I was going to ask you about it.

Steve: Yeah, it was worthwhile. The premise was that it was known that our security industry, our security infrastructure understood that it was possible to factor primes. So they didn't want it to be made public, so they were spying on all the top mathematicians who were working in the field that might stumble upon this. And so anyway, it was, you know, I mean, again, that's where I was thinking, wow, they got a lot of this right.

Leo: I was ready to lambast them. I thought, this is going to be terrible. But good, that's good to know.

Steve: Well, there were some things that were not correct.

Leo: Okay.

Steve: They didn't actually say - it wasn't factorization, but it was primes. They understood that something about primes.

Leo: They understood primes were important.

Steve: Something about primes. Oh, it was patterns in primes. It was some guy was like, oh, he, like, figured out like a pattern in primes. But it turns out that this - so this was a conspiracy to keep this from being...

Leo: Keep it quiet.

Steve: ...discovered, to keep it quiet, that went back decades. And so anyway, it was - I would say it was fun.

Leo: And you and I both watched Robert De Niro's "Zero Day," which also had some technical accuracy in it. So, you know. They're getting better. Anyway, time to take a break. Nothing but technical accuracy just around the corner, and our Picture of the Week. But first...

Steve: Also technically accurate.

Leo: Yes. Is it? Oh, good.

Steve: Uh-huh.

Leo: I haven't looked yet. All right. I like to save it for the show. All right. Let us go back to the show and Steve Gibson's Picture...

Steve: You know, those .security domains cost $2,500 a year.

Leo: Oh, you looked at it, I bet, yeah.

Steve: And I don't think that's in keeping with the founders' intent for the way the Internet would work.

Leo: That's expensive. Yeah, these custom emails really, you know. But on the other hand, they're nice. Like I have Leo.pizza. And I think if you wanted to...

Steve: Well.

Leo: All right. Let's look at the Picture of the Week, Mr. Gibson. I'll scroll up here.

Steve: This one I gave the caption "Making the Switch from Windows to Linux."

Leo: I'm trying to understand it.

Steve: Apropos of last week's podcast about the EU OS.

Leo: If you scroll all the way up you get it a little bit better. Okay. Broken telephone pole.

Steve: Ah, yes. And again, this just - these pictures beg so many questions. So for those who can't see, we've got a buckling, broken telephone pole that some hapless lineman has tried to keep erect with duct tape.

Leo: Oh, my god, duct tape's keeping the world together, yes.

Steve: It looked like maybe there's some sticks on the outer side that were used...

Leo: Like splints?

Steve: Some splints, exactly. So it was, like, splinted, and then duct taped - the splints were duct-taped to the pole, just trying to keep it up. But then over to the right we see the one that I've labeled Linux which has like a new pile of dirt at its foot.

Leo: It's the replacement pole, clearly.

Steve: Exactly. It must be the - and then I don't know why there's little rope strung between the two.

Leo: That's the funniest thing. I don't understand that, either.

Steve: It's like a leash. It's like, don't go away, boy. Stick close.

Leo: That is the funniest thing ever. Duct tape, man, it holds the world together.

Steve: It is, yes, exactly.

Leo: Wow. And Windows is the duct-taped solution, and of course the brand new, perfectly formed pole is Linux.

Steve: That's right.

Leo: I like that, Steve. Thank you.

Steve: That's right. It's one of the expressions we have around the house when one of us wakes up and something is stiff. We say, oh, get the duct tape.

Leo: Really. Okay. I'm not sure, that was maybe a little too much information. That's good.

Steve: Oh, like a stiff muscle is what I meant.

Leo: Oh, sore.

Steve: Like a shoulder.

Leo: Sore, yes, Steve.

Steve: Like a stiff shoulder. Sorry. Whoops.

Leo: Of course. Get the duct tape. You never know.

Steve: That's right.

Leo: I have a vision of you duct-taped to the bed. Okay. So maybe that's not...

Steve: Okay. So the Microsoft Offensive Research and Security Engineering - this is one of those reverse engineered acronyms. The abbreviation is MORSE, M-O-R-S-E, Microsoft Offensive Research and Security Engineering. They've identified a crucial security vulnerability within a range of Canon printer drivers - Canon, you know being a leading, very popular printer - which threatens users across, well, anybody who's using that printer who would be a target. The vulnerability could reportedly allow malicious actors to compromise printing operations and, in severe cases, execute arbitrary code on affected systems.

We did a podcast years ago that I thought was one of our better ones, where we looked at the threat that something as innocuous-seeming as a network printer in an enterprise could pose, because it was discovered that Advanced Persistent Threat actors were actually setting up shop in enterprise's printers, which were not being scanned. You know, they didn't have Windows Defender running on them. It was just a printer. But it

turns out, you know, it's a computer, probably running Linux of some flavor. And they were able to just stay ensconced inside this printer for quite some time.

Anyway, this has a CVSS - the concept of a printer driver in this case, not the printer itself, but the printer driver in a Windows system - has a CVSS score of 9.4. As we know, that's a high-severity risk. That's up at the high end of the scale. And it has a 9.4 due to its lack of complexity. Very easy to leverage the bugs in these Canon printer drivers. You do not need any elevated privileges to use it, nor any user interaction. The potential for high-impact compromise of confidentiality is there. So 9.4. It provides a path to deliberate memory corruption during the EMF Recode processing, which is something that the printer driver does. Probably EMF is Enhanced Metafile, I'm pretty sure. And unfortunately, this opens systems that do not use Canon printers to the infamous BYOVD attacks, where BYOVD is short for "Bring Your Own Vulnerable Driver."

The problem is, these vulnerable Canon printer drivers were originally signed by Microsoft. You know, Microsoft blessed them, allowing them to then to be loaded without a second thought into Windows. So they can't be altered at all, or that would break the signature, and then Windows would refuse to load the driver into the kernel. No need to alter the driver because it's buggy, and now the bugs have been found. So malware can bring along one of these flawed Canon printer drivers, drop it onto the system, get it loaded into the kernel, and then leverage the flaw in order to take over the system.

When an entity has Canon printers, they're there by default, across a variety of printers including their production models, home and office automation multifunction printers, and laser printers. So all that a malicious application needs to do is cause a print job to be processed through the vulnerable driver. That allows the attacker to gain control, you know, and have kernel-level access, which is to say root on the system.

Canon has acknowledged the issue and has promised to be releasing updated drivers as soon as they can. So if you are a Canon user, that means your system already has these vulnerable drivers in it. And, you know, the malware doesn't need to bring its own along. So keep an eye out for any updates that the Canon offers. You'll certainly want to make sure that you are receiving Canon's notifications of updates. And I imagine that what will happen as soon as the new drivers are present, and given some opportunity for them to filter out into the ecosystem, is that before long Windows Defender and the other endpoint management, you know, third-party software will start explicitly looking for these known vulnerable drivers and say you really don't want to be loading this.

And that's the way the Bring Your Own Vulnerable Driver problem will get resolved is that, as soon as replacements are available, so that functionality isn't killed when the vulnerable driver is removed, then those drivers will just be blacklisted, and you won't be able to load them into Windows anymore. So all this takes time. And as we know, everything now is an arms race to see how much infiltration and how much damage can be done before the problem is resolved.

Okay. I've talked about an astonishing home appliance company. This was thanks to a piece of feedback that we received from listener David Morrell. David wrote: "Every home IoT device maker should follow the lead of this home appliance maker. About the only thing" - and Leo, I have to say when I was looking at the site, I thought, oh, these look like appliances Leo would want. I mean, they are really beautiful.

Leo: Don't tempt me, Steve.

Steve: The company is Fisher Paykel. Yup, you've got it up on the site now. He said: "About the only thing they could have added is advice to use a YubiKey or similar."

Meaning they really get it. And he said: "They really get it." He said: "And it even looks like you can buy these" - oh, they're New Zealand. "You can buy these New Zealand-made home appliances in the U.S. Personally, I'm quite happy not having IoT in my home appliances."

Okay. So David's note made me curious. I went over to the Fisher Paykel website. It's F-I-S-H-E-R P-A-Y-K-E-L dotcom. And I discovered that they have an entire page devoted to the cybersecurity of their well-connected appliances.

So to give everyone a sense for what's there on this home appliance maker's site, 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." The fact that they even know the term "core security values."

Leo: They use WPA3. That's all I needed to see. It's like, wow, wow.

Steve: Like, astonishing. It's like, and I have to say...

Leo: There's a geek in there somewhere.

Steve: I have to say, Leo, I wonder if someone's going to be smiling when he hears me reading this because he's a Security Now! podcast listener. I mean, because, like, I mean, some guy at Fisher Paykel because...

Leo: It feels like it, doesn't it. It's everything you would say.

Steve: It sounds like the guy's been listening to us. The page says: "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."

He said: "Security by Design: Security controls to protect appliance data, user authentication and authorization, 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" - who talks about the coding of their dishwasher? - "and is developed by the Test-Driven Development software method."

Leo: Yeah, right on.

Steve: I mean, the guys must have like some nephew who's into serious security or something. This is just amazing. They said: "Any third-party and open-source software is analyzed 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." I mean, I almost want to buy this stuff just to support these people. It's amazing.

They said under 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 WiFi router using the WPA3 network security protocol as standard, with WPA2 for backwards compatibility. The appliance does this even if your router is not set to this configuration. That's just one example..."

Leo: So awesome.

Steve: "...of how Security by Default is engineered into your appliance." And then, 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." I wish these guys made, like, some social networking software because we could give it to our government, and it would be way more secure than what they're using.

Leo: Wow.

Steve: They said: "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."

And for 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."

Leo: They look pretty darn good, you're right, Steve. And you can buy them in the U.S.

Steve: Oh, they're gorgeous. I mean, the equipment is beautiful, Leo.

Leo: Yeah, yeah.

Steve: I mean, the people who did the industrial design are friends with the people who did the security design. I mean, it is topnotch.

Leo: Look at that, yeah. Also top prices. $15,000 for an oven.

Steve: Oh, but honey, it'll sing you to sleep.

Leo: I have Internet connectivity on my oven.

Steve: Of course you do.

Leo: The only value at all is it will tell you when the oven's preheated on your phone, say "Hey, your oven's ready."

Steve: Go put your roast in.

Leo: Go put your roast in. It's ready. That's it. Door's open.

Steve: Industry leaders in IoT security. Actually, we could use that for our refrigerator. Luckily, our refrigerator sounds an alarm.

Leo: Beeps at you.

Steve: Lorrie just walks away. I don't know what's going on, but like, honey, you know, not only are the lights on, but the refrigerator's open.

Leo: Yeah, yeah, I've done that, yeah.

Steve: So anyway: "Reach out to us," they said, "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." It's just astonishing.

And then they said, under Our Ratings: "We are proud to have achieved the Gold verification level for UL's (Underwriter Laboratories) IoT Security Rating." I didn't know Underwriter Laboratories did 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 is an oven, folks. This is not like a server or a router or, you know, or an endpoint security device. This is somebody's microwave. It's just...

Leo: Unbelievable.

Steve: ...astonishing. So anyway, props and a salute to FisherPaykel.com. And if anyone from there is listening to this podcast, congratulations. Oh, and Leo: "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."

Leo: Holy cow. Wow.

Steve: "At is@fisherpaykel.com." And here it comes. "Note: We support PGP encryption using the Fisher & Paykel Appliances Information Security PGP Key."

Leo: All right. Now I'm going to give you the bad news.

Steve: Oh.

Leo: It's a subsidiary of Haier, which is a giant Chinese multinational. So, I mean, you know, maybe they could spread the word throughout the entire Haier ecosystem.

Steve: I wonder if they - they probably use open source, but don't publish their firmware.

Leo: Yeah. I mean, I think, you know, every - nowadays every - this company was acquired, obviously.

Steve: Yeah.

Leo: Haier's a giant monster conglomerate.

Steve: Right, so they just sucked them up because they said these guys are doing it right. We want...

Leo: And they have got a high-end brand, right, yeah.

Steve: We want a piece of their action. Boy, it's beautiful.

Leo: Just as they have low-end brands, yeah.

Steve: Oh, and get this, Leo. I just - I couldn't imagine. After all that, they then have sort of an FAQ Q&A thing where they talk about to their customers how to enhance their security.

Leo: Wow.

Steve: And they finish with "Separate Networks: Security experts recommend creating separate and secure networks dedicated for your IoT devices" - which makes me think, are they listening to this podcast? - "that separate from your network used for banking or ecommerce 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: using one router and setting up a 'guest access' or a 'guest network' within the router settings; or use separate routers paired with your Internet connection."

Leo: Oh, they definitely listen to this show.

Steve: Incredible. "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.'" Anyway, I am 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 best imaginable April Fools Spoof, since no one would ever believe that I hadn't made this entire thing up from scratch. And Leo, if the rest of the world designed and built their equipment like these guys, it feels as though our job here would be done.

Leo: That's impressive. I wish all, yeah, I wish all IoT stuff was like this. That's incredible.

Steve: Incredible.

Leo: Yeah. All right, Steve.

Steve: Okay. So 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 which instead redirected any students who were foolish enough to click on it to a phishing awareness video. Now, what was interesting was, according to France's privacy watchdog, over 210,000 students did click the link, but that's only one in 12 students out of a population of 2.5 million.

Leo: Impressive.

Steve: Yes, 8%. And, you know, while, yes, 210,000 is a lot of individual students, they fared way better than the one-third click rate which is typically seen in corporate environments. So the old folks in the corporations, eh, like, oh, I can get free socks for life? Great. But, you know, these kids are like, eh, I don't think so. This looks like junk. So congratulations.

As we've observed before, with 521 million websites built on WordPress - 521 million.

Leo: That's mindboggling, yes.

Steve: It is. It's like...

Leo: It's almost half.

Steve: ...43.5% of all websites in the world are WordPress. So its security, WordPress's security, is always a top concern. So much of the Internet depends upon it. So when, three years ago, in 2022, 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," It's like, what? What could possibly go wrong?, which is, you know, our rhetorical question.

They said, this is how they described this feature: "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 they 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 would break the site. Must-use plugins are always on, with no need to enable via admin, and users cannot disable them by accident. They're enabled simply by uploading a file to the mu-plugins directory, without having to log in," even.

This, of course, as I said, is where we cue one of our favorite rhetorical questions: "WHAT could POSSIBLY go wrong?" Yes, you just have the file there, and WordPress won't show it to the admin, won't require you to be logged in to enable it. In fact, you can't enable it. It's always enabled. And you can't disable it because they said, well, it would break the site if this plugin wasn't there, so we're just going to, if it's present in this directory, run it. GoDaddy's Sucuri security team provides the answer to the question about what could possibly go wrong, and unfortunately that's not rhetorical.

To no one's surprise - except I suppose the creators of this very abuse-prone feature, I mean, they must be surprised, but, like, duh - hackers are now abusing this little-known WordPress feature to install and hide their malware from site admins. According to GoDaddy's team, threat actors have been found to be abusing, to no one's surprise, Must-Use Plugins since at least February of this year. And that abuse has recently grown worse. It's like, hey, this works. Let's use it everywhere.

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 the site backend management. As an added benefit, because it's a relatively unknown and under-the-radar feature, many WordPress security tools don't even scan the mu-plugins folder for threats. They're not even looking. Sucuri has seen attackers use 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 admins are advised to keep a watch on the content of that folder. If it's currently empty, unused and unneeded, delete it entirely and make sure it stays deleted. So stepping back from all this, it appears that the design of this makes it far too easy to both use and abuse. With a design like this, it's not possible to have ease of use without also inviting ease of abuse. So again, to our listeners, given that more than 500 million sites, or more than 43% of the Internet is WordPress, it must be that our listeners, that a big chunk of our listeners are affiliated one way or the other with sites that are being run by WordPress.

So take a check. It's under the wp-content directory, the default content directory, mu-plugins. It's probably empty. WordPress brought it along for the last three years, since 2022. It's more than likely, whatever your host is, it doesn't need it; but it's there waiting to be abused. First of all, make sure that, if there's anything in there that you know what it is and why it's there, get rid of it and get rid of its directory, if you don't know that you need it, because this is under active exploitation. You know, they do have to break in somehow first. But achieving persistence or planting malware somewhere where it won't be found and quickly discovered is the second part of the challenge. And if it's a WordPress-based site, and the mu-plugins directory is there just waiting to run something that you drop in, that's what the bad guys are going to do. Meanwhile...

Leo: Meanwhile.

Steve: Meanwhile. Oracle, the massive organization with designs on running TikTok, although I thought that was interesting, Leo - by the way, we should mention that on Sunday's TWiT show you had Jason Calacanis, who - he's a great guest. You've had him through the years.

Leo: Yeah, he's an old friend, yeah.

Steve: He's an old friend. Super smart guy. And he happened to mention, the thing that made me think of it is he was thinking that Amazon, right, wasn't that what Jason thought?

Leo: He said Amazon's going to be the TikTok - yeah. We'll see.

Steve: Manager. We'll see. What we'd heard was that it was going to be Oracle, that down in Texas, you know, the big database company, and they were going to be, you know, managing TikTok and retaining TikTok's U.S. domestic data. Anyway, whether or not that happens, whether it's Oracle or Amazon, and TikTok just got another 75-day extension, right, because the boom was about to be lowered on it again.

Leo: Yeah. Yeah. Yeah, Saturday was the deadline.

Steve: Yeah. Okay. So Oracle 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 well back at the end of January, and the hackers are using the stolen data to extort U.S. medical providers. So this is not apocryphal. This actually happened. Yet Oracle has said nothing. They've made no report of any breach, as is required by law, to the U.S. 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 single-sign-on login servers and the theft of account data for six 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 six million users. The threat actor also said that stolen single-sign-on 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," wrote Lawrence, "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," meaning Oracle 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.' He said: 'This denial, however, contradicts findings from BleepingComputer, which received additional samples of the leaked data from the threat actor and contacted the associated companies.'" Bleeping Computer reached out to the affected companies. "'Representatives from these companies, all who agreed to confirm the data under 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 had hacked Oracle's servers. 'I've dug into your cloud dashboard infrastructure and found a massive vulnerability that has handed me full access to info on six 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 Proton 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 an @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 [S-E-K] has also found an Archive.org URL showing that the login.us2.oraclecloud.com server was running Oracle Fusion Middleware 11g as of February 17th 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."

So 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 mandates that publicly traded 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; and 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 has been publicly disclosed.

And I would argue, I mean, you know, Lawrence did a beautiful job of really pursuing these facts and essentially demonstrating proof of a material breach. And the fact that they had a server running known buggy and patched four years ago authentication frontend, and the attacker said that's the bug they used to get in, and now that server is gone, I mean, it seems like an open-and-shut case. And Oracle is really misbehaving badly. So for what it's worth. Unfortunately, their lack of responsibility-taking is exposing the authentication credentials for six million people who trust them. So it's not like this is nothing. This is not good, and those six million authentication credentials are now for sale on the dark web. And apparently there's a means of decrypting them using information that the attacker also has.

So, you know, this is not just Oracle choosing not to say anything because they don't want to affect their stock valuation. It's also materially hurting their customers. I mean, this is, you know, a class-action lawsuit against them pending. It's hard to see how it wouldn't be, not that you or I are in favor of that. But, you know, they need to take responsibility.

Meanwhile, I wanted to note that nearly two weeks ago, as we mentioned two weeks ago, that Utah law we talked about which had passed through their legislature was now signed into law by Utah's Governor Spencer Cox. Formally known as the App Store Accountability Act, or S.B. 142, the new law mostly takes effect a little over one year from now. So as always when, you know, some new law goes into effect, it is going to require a significant change in behavior, then a period of time, you know, a grace period, that's the word I was looking for, a grace period is part of the law to allow people to get themselves ready. That occurs on May 6th of 2026, given that the law stays in effect until that time.

It's on May 6th, 2026, a little over a year from now, that the law's core requirements, including age verification and parental consent mandates, will take effect. So that'll give, you know, the app stores, developers, regulators time to prepare for coming into compliance with these new regulations. And of course it will give other states time to decide if they want to follow suit.

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. The 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 as those who run the stores, and away from the individual apps like Instagram, Snapchat, and X to do the age checks.

This does beg the question, though, what about apps that are already downloaded and installed from app stores when May 6th rolls around next year? Are those grandfathered in because they're already there, and they're allowed to stay without verification? Or will they need to then be reverified? Don't know. Regardless, the passage of this App Store Accountability Act is expected to trigger something. South Carolina and California have both been rattling their sabers, saying that they're looking into doing this. 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 are unable to agree to them meaningfully. Todd Weiler said: "For the past decade or longer, Instagram has rated itself as friendly for 12 year olds." He says: "It's not."

So the Utah law is expected to face legal challenges in fights over its validity; but, as we know, my own take on that, on this whole thing, is that, yes, in cyberspace something needs to be done. If we're going to decide that children's age matters, then responsibility needs to be taken somehow. And I think that the most recent begrudging proposals that have been made by Apple and Google make the most 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.

And that solves the problem. The apps don't obtain any information about the ages of their users, and the devices are responsible for getting permission if they've been configured to require it. So, you know, Apple and Google have both articulated that solution, and I imagine that we're going to see that happen. And that'll be good, and not a huge loss of privacy.

This was an interesting piece, and I guess you saw that, Leo. It turns out that AI bots are inadvertently DDoSing FOSS, you know, Free and Open Source Software repositories, in their endless quest for more publicly available content.

Leo: Yeah. Wikipedia's been complaining about this. It's a real problem for them.

Steve: Yeah. Oh, Wikipedia has.

Leo: Yeah. Think about it.

Steve: I guess that means...

Leo: Yeah, Wikipedia's a great resource for that.

Steve: And they want to be a public, I mean, they want to not restrict themselves in any way. They want to be a public resource. Wow. So Ars Technica did a great job of reporting on this worrisome trend that's been developing and worsening through the year. They said: "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 their user-agent strings and cycling through residential IP addresses, using them as proxies. So, you know, actively working to avoid being blocked.

"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." Basically, you know, proof of work in the browser, you know, again, solve computational puzzles. So spend time per access, per query, to validate themselves. We've probably run across this on Cloudflare. Sometimes you'll come to a Cloudflare page where it'll just sort of hold you for a while, while something appears to be going on. And that is typically a proof-of-work, you know, requiring some script in your browser to do some heavy lifting which no high-rate bot is able to afford because every single time the bot tries to access, it's hit with this barrier to entry, essentially.

So Ars wrote that Iaso had written in a blog post titled "A desperate cry for help," he said: "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," they wrote, "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 attacks on vital public resources. According to a comprehensive recent report from LibreNews, some open source projects now see as much" - get this - "as 97% of their traffic originating from AI company bots" - 97% are just bots trolling - "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% of requests, that's 2,690 requests 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." Naturally. "When many people access the same link simultaneously, such as when a GitLab link is shared in a chatroom, site visitors can face significant delays." Ah, so something triggers that challenge, like when there's enough repeated access to a link, that suddenly switches on the challenge, which is not always on there all the time otherwise. So they said: "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% 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%, 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 agent strings, 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 configuration 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.'" Meaning a false positive, actually a true positive detect on a large company's IP address block because they just had to shut down all access to that company to their site because their blocklist has become so large.

"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, crawlers often hit expensive endpoints, like git blame and log pages, placing additional strain on already limited resources." And by that they're talking about an expensive endpoint is some page which requires a lot of database access or backend work in order to produce the page. And so if the robot just hits that continuously, it's very resource expensive in terms of computation and access 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." Right? You know, to track them down and realize this isn't - what is this? It's not an actual vulnerability.

"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% and Anthropic for 4.3%.

"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 six hours because 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 model knowledges 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 page 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 the 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.'" Okay, I'm not quite sure how that's that different from Nepenthes. "'Cloudflare reported that AI crawlers generate over 50 billion requests [wow] to their network daily. AI crawlers generate over 50 billion requests to their network daily, accounting for nearly 1% of all web traffic they process.'" Which says they're handling, what, 5,000 billion requests? Yeah, 5,000. So...

Leo: Five trillion.

Steve: Yeah, five trillion. Five trillion requests per day. Wow, Cloudflare. "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."

Leo: Yeah, they should honor those. That's key; right? Yeah.

Steve: Yes, yes, exactly. "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" - and I would add, and deliberately ignoring their clearly established standards for saying please don't - "risks severely damaging the very digital ecosystem on which these AI models depend."

And finally they wrote: "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." Yeah.

Leo: Yeah. If they don't honor robots.txt, then anything you do to them is fine.

Steve: Right. If they're - exactly. If they're deliberate - that's a very good point, Leo.

Leo: Yeah.

Steve: If, you know, we might say, hey, it's kind of foul play, sending them into an AI-driven tarpit. But if you first said "Don't go in here because of what's in the robot.txt..."

Leo: Right. Exactly.

Steve: And I presume they do.

Leo: Cloudflare does do that.

Steve: Yes.

Leo: Yes. By the way, Nepenthes is funny. So Cloudflare calls it a tarpit. But a Nepenthes is a pitcher plant. It's the plant that traps bugs.

Steve: Oh, right. That, like, the...

Leo: It's not a Venus fly trap. It's a pitcher. It has dew in it, and the bugs move into it, and then of course it eats them. So it's just like a tarpit, but it's a plant version.

Steve: Very nice. Very nice.

Leo: From the plant kingdom of a tarpit. I think that's very funny, yeah. All right. Back to you, Mr. G.

Steve: So 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 that they've made, under the section "Other," Microsoft wrote, and I love the way they phrased this: "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."

So, okay. 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 convenience or experience. But that's, you know, what will henceforth be required for all users setting up Windows 11. And 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 may be required. But it is a change that I wanted to let our listeners know about.

Some of the reporting I saw about this phrased it a little differently. They 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." So anyway, having shared all that, I won't be surprised if there isn't soon a workaround for this, we've seen those before, when this has sort of been there.

Leo: It's actually a little more - it's simpler than this. We talked about this on Windows Weekly, which is how I know.

Steve: Oh.

Leo: That was a script, a powershell script, actually maybe not even a powershell script, there was a shell script that launched a series of commands. Those commands are still there. And so what Microsoft has done is make it so that somebody who is non-sophisticated won't have a simple, oh, just click this and it'll run, the bypassnro script. But all of the commands that do bypass the Microsoft login are still there. They have not removed those. So Paul's position on this is you still can set up Windows 11 without a Microsoft account. But you need to be a little more sophisticated than you used to be. And that's Microsoft's intent because, for instance, if you're using Windows Home, it turns on BitLocker, but only if you turn on your Microsoft account because you need a way to store that certificate. So many people lose their certificates.

Steve: Right.

Leo: So Microsoft's erring for the - I think this is, I've always said this is the ideal solution, which is - and Apple does this, too.

Steve: Kind of a way around it.

Leo: Yeah. By default you make it more secure, but less flexible. But if you're in the know, if you're a sophisticated user, there are ways to disable it.

Steve: So they took it out of the GUI.

Leo: Basically.

Steve: That little skip for now or local account that they used to have.

Leo: Right. But Paul says, at least for now, and he believes this will continue, it is absolutely possible to do this. You just don't have that script to do it anymore.

Steve: Well, and it's...

Leo: But if you look in bypassnro.cmd, you could see the commands. It was just...

Steve: Well, and it would seem to me that even if you - they wouldn't remove the ability to have a local account. So even if you had to temporarily create a Microsoft account to get installed, then you add a local account and delete the Microsoft account.

Leo: That's what Paul's recommended workaround is. You know, you can make a dummy Microsoft account that you don't use.

Steve: Right, that's just to get you installed.

Leo: Exactly.

Steve: And then, yeah, and then...

Leo: And they can't get rid of that. As long as there is a local login at some point, yeah.

Steve: Right.

Leo: So I think it's not, just as you say, you said that you wouldn't be surprised if there's a workaround. There is, basically, and they're never - they didn't get rid of that. Yet.

Steve: Okay.

Leo: Yet.

Steve: Well, and again, as I said, I don't mean to make a big deal about it. You know, it's just annoying to be constantly asked if you want to - you haven't backed up your drive. It's like, hey, I've got my own backup. You know, there's no way to tell it to shut up.

Leo: It's not for you, it's for normal users.

Steve: Yeah.

Leo: That's the problem. And it's always been the challenge in technology to make it reliable and safe for normal people, but to give us hardcore users the power that we really want. And deserve, yeah.

Steve: Okay. So I love this. 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 web browser, where it's then decrypted. So, technically, yeah, end-to-end. But otherwise, Google jumped through some weird hoops to offer this.

Okay, now, since the technology is interesting, and since it might well be of interest to our listeners whose corporations might find value here - because, I mean, it's not

nothing. It's just not really, you know, what we're used to. I want to take us into the details. And for that, Ars Technica's Dan Goodin did a terrific job of setting this up, creating the appropriate context, and explaining what goes on. Ars's headline last week about this 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."

So Dan opens by saying: "When Google announced Tuesday that end-to-end encrypted messages were coming to Gmail for business users, some people balked, noting that it wasn't true end-to-end encryption, 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'm going to interrupt here just for a moment to note that the way the conventional end-to-end encryption operates is pretty straightforward. So let me set that context first because he doesn't do that. Each party, as we know, 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 the message, anything she wants. That's the so-called "bulk encryption" key. And that's just randomly, you know, she creates a high-entropy random secret symmetric key which she uses to encrypt her stuff. She uses that 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, the recipient's, public key. She then packages the encrypted message up along with the result of the double key encryption and sends that package to Bob. Upon receiving Alice's package, Bob first decrypts the double-encrypted key using his secret key, which undoes the second encryption that Alice put on which used Bob's private key. And of course only Bob knows his private key. He then looks up Alice's publicly published public key and uses it to decrypt the result of the first decryption. And the beauty of this is that only if all four of these keys were correct will Bob now have recovered the properly decrypted secret symmetric key, which he can then use to decrypt the package that Alice prepared for him.

Now, 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 parties' private keys must be used, and only each party knows their own private key, not only do we get strong encryption protection from anyone attempting to intercept that communication, but Alice knows that only Bob can decrypt what she encrypted, and Bob knows that only Alice can have sent what he decrypted as having come from her. So that's true end-to-end encryption, and that's not what we got from Google in Gmail.

Okay. So Dan explains what we did get. He wrote: "When Google uses the term end-to-end encryption 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 cannot 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." So in other words, they sort of skinned the cat here in a different way. They've come up with something that complies with the regulations for end-to-end encryption, yet made it much easier to deploy.

Dan said: "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 end-to-end encryption 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." Okay.

"To make this happen, Bob's organization deploys what Google calls a 'lightweight key server,' known as a KACL, short for 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." Thus ephemeral.

"To prevent Mallory or another adversary-in-the-middle" - Mallory-in-the-middle - "from obtaining the key, Alice must first authenticate herself through Okta, Ping, or whatever other Industry Identity Provider, or IDP, Bob's organization uses." So Alice must authenticate herself to Bob's organization's identity provider. Dan said: "If this is the first time Alice has received a message from Bob's organization, she'll 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.'"

Okay, now, I'm going to interrupt here again to note that in no way is any web browser a safe place to decrypt super-secure, you know, like national security level or extremely proprietary corporate material. You know, like in the same way when we were talking about Signalgate, as it's now being called, of national security-level secrets being transacted on people's individual smartphones, it's not Signal that had a problem because it's true end-to-end encryption. It's that it's on the smartphone device. It is decrypted after it arrives. So we have the same problem with a web browser; right? You know, 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 zero-day style security vulnerabilities. That's still happening.

You know, if you really need to send something securely, my advice would be encrypt it offline, away from any web browser, then send it in the clear through any email system. Doesn't matter because it's been, you know, it's PIE, Pre-Internet Encryption. Pre-web browser encryption. You know, and 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, which is today's web browser, that any computer system has today.

Dan finishes his description by writing: "Now, as to whether this constitutes true end-to-end encryption, it likely doesn't, at least under stricter definitions than are commonly used. To purists, end-to-end encryption 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 end-to-end encryption. 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 Client-Side Encryption. 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."

So anyway, if anyone was wondering, you know, heard about Google's, you know, end-to-end encryption, now we have some context. It's certainly better than what they had before. If your organization wants to use it, then, you know, it does keep things encrypted. But, you know, if you're using Gmail in your browser, you have an HTTPS connection to Gmail.

Leo: Right, right. And anything that goes Gmail to Gmail remains encrypted.

Steve: Yeah. It's never been in the clear at any point.

Leo: Right, right. I think this is really for businesses that don't want to give up full encryption, right, because they want to make sure that they can monitor your emails. In fact, they may have a regulatory requirement.

Steve: I think it's an interesting regulatory hack.

Leo: Yeah.

Steve: I think that's it. I think it's, you know, it's like Google was under some pressure to come up with a way for regulations that require end-to-end encryption, like the letter of the law, that it's encrypted on your device, decrypted on the recipient's device. And Google said, oh, yeah, we can do that.

Leo: Did you ever wonder who Bob and Alice are?

Steve: I do. And boy, they have some longevity. They're still talking.

Leo: Sometimes there is a Ted and a Carol that gets involved in these conversations.

Steve: Yeah.

Leo: And it all comes from a 1969 movie about wife-swapping called "Bob & Carol & Ted & Alice."

Steve: Ted and Alice.

Leo: You remember that; right? Yeah.

Steve: Yup. We're older. We've been around long enough.

Leo: Yeah, us oldsters know where that came from. It's pretty funny. And I would imagine people listening who don't know that are going, who are these Bob and Alice that everybody's always talking about when it comes to encryption. I think that's where it came from. It seems like a coincidence if it didn't.

Steve: Must be. And it has the advantage of having A, B, and C - Alice, Bob, and Carol.

Leo: Yeah. Ted we just can throw out. We don't...

Steve: Yeah, Ted, you know.

Leo: He doesn't fit.

Steve: And then Mallory as Mallory-in-the-middle.

Leo: Oh, there you go.

Steve: Mallory is also the name used for your attacker.

Leo: For man in the middle. Oh, nice. That's nice. Do you want to pause, or do you want to keep going? We've got time.

Steve: I've got a little bit more, and then we've got some - oh, yeah. One more, and then feedback, when we will pause.

Leo: Okay.

Steve: So, but this is an important one for anyone who is running Apache Parquet. A CVSS 10.0, which we know is very difficult to achieve. It's like the Olympics of bad vulnerabilities. Apache recently received the much-dreaded full CVSS 10.0 with a widely used module known as Apache Parquet, which is spelled P-A-R-Q-U-E-T. Apache Parquet is an open-source, columnar - as in, instead of rows, it's columns. So columnar storage format designed for more efficient data processing. Unlike row-based formats such as CSV, Parquet stores data by columns, which makes it faster and more space-efficient for analytical workloads. It's 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 Apache Parquet system.

Leo: Wow.

Steve: Yeah. Unfortunately...

Leo: How widespread is Parquet use? Is it a pretty popular...

Steve: Among those who use it. I mean, Netflix, Uber, Airbnb, LinkedIn.

Leo: So, okay, yeah.

Steve: I mean, Hadoop, AWS, Amazon, Google, Azure cloud services. So, yeah.

Leo: Okay, yeah.

Steve: It's got some wings there. Unfortunately, the problem was disclosed on April 1st. But since this is no joke, and it would be horrible for those affected if they thought it was, I hope no one dismissed it as an April Fools event. This maximum severity remote code execution problem impacts all versions of Parquet up to and including 1.15.0. The problem stems from the - here it is - the deserialization, we've talked about deserialization flaws because they're tough, of untrusted data. And of course deserialization is also known as "interpretation."

And we know how hard it is to do interpretation correctly. It could allow attackers with specially crafted Parquet files to gain total 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. It was fixed with the release of Apache version 1.15.1. So it is some solace that in order to exploit this flaw, threat actors must convince someone to import a specially crafted Parquet file for Parquet to then deserialize. But we all know that social engineering attacks remain some of the hardest to defeat. And it might well be that there are other vectors.

So anyway, I wanted to put it on everyone's radar. If you happen to know that you're using Parquet or know someone that does, the good news is it has not been publicly

leveraged. It's not known to be used. It was discovered by Amazon AWS security folks because AWS uses it. They told Apache. Apache's updated it. But we know how that goes. So the bad guys will look at new and old Apache, do a dif of it, see what's changed, reverse engineer the exploit, and then go looking for publicly exposed Parquet instances. So if you're using Parquet, update immediately because you want to beat the bad guys to it. And now, Leo, let's take a break.

Leo: Butter. Oh, I thought you wanted me to say "butter." Okay.

Steve: Butter, butter. Parquet.

Leo: Butter.

Steve: Parquet.

Leo: All right. Back to Steve.

Steve: So @TechnoAgorist, so this must have been through X where I checked in, he wrote: "Regarding Neal Asher's novels, they may not be on Kindle Unlimited, but I found them at my local library."

Leo: Nice.

Steve: "That's how I've been reading them. Thanks for the recommendation." And I appreciated being able to share a reminder about printed books.

Leo: Yeah, nothing wrong with them.

Steve: Yeah. I'm still enjoying Neal. I'm Book #4 of the first five-book Agent Cormac series, as it's called, and I'm having a great time. The books are long and involved. The style Neal uses for the first three, at least - and yeah, I guess it's to a lesser degree now in Number 4 - was to create several parallel plot lines that initially don't appear to bear any connection to each other. There's no obvious relationship. So you'd sort of move around between them, and you're thinking, okay, why do I care about this person? But, you know, as the story progresses, they eventually converge, and you end up - I remember at one point thinking I'm having a lot of fun with this book. So anyway, thank you to TechnoAgorist for your note about books are still available from libraries in print.

Leo: Amazing.

Steve: That's certainly a way to get it.

Leo: Who'd a thunk it?

Steve: It wouldn't occur to me, Leo, I have to tell you. Eric Seidel said: "Hey, Steve. 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 two-factor authentication reset requests show up in my email, and then happened to look in my sign-in activity. And it is a sign-in request every minute to my account. It's just insane. Make sure you have your two-factor authentication turned on. Holy smokes."

And I put in the show notes just a snapshot that he had sent me that does, you know, indeed show, in fact, sign-ons like every minute or several times in the same minute. So again...

Leo: All this because they didn't have the 2FA code.

Steve: Yeah, the idea that some guy, I mean, or that, you know, the bots apparently are just sitting here...

Leo: Amazing. Hammering it.

Steve: ...pounding on people's email without better protection. And it is really disturbing. Matthew West said: "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 in order to pair it. I'm trying to pair by the constantly changing one-time code in the hopes it eventually works." In other words, he's guessing.

Leo: Oh, forget about it.

Steve: He said: "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."

Well, Matthew, we previously addressed this question a few months back, when we took a deep dive into the precise operation of hash-based one-time passwords. That podcast was 1009, and we received an unusual amount of positive feedback from our listeners...

Leo: Yeah, it's great.

Steve: ...who enjoyed thinking about the various aspects of a six-digit code that was changing randomly every 30 seconds. The answer to the first part of your question, Matthew, what's the best strategy, is that since the proper PIN code at any given instant is 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 the best strategy because a great deal of that is going to be necessary.

So exactly how much? The second part of your question asked how many attempts would be needed to reach a 50% chance? And that is 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, and that is the case, if the probability of a correct guess is one in a

million, since it's from 000000 to 999999, that's a million possible combinations. We might tend to assume that giving that one-in-a-million thing one million opportunities to occur - or in our case one million guesses - we would probably obtain a collision of six-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 is a 63.21% chance of hitting it. So, you know, better than 50/50." Okay. "For random events, it's all about probabilities."

And so here's the answer to your question, Matthew. 693,147 guesses, so just shy of 700,000, would be required to hit the 50/50 point, for an even chance of any of those one-in-a-million guesses being correct. So that's why patience will be the best strategy. Maybe getting a different Fitbit would be a better idea because you're going to be guessing for - I don't know how fast you can guess, but it's going to take just shy of 900,000 guesses to reach the 50% point. That would try my own patience.

Leo: Yes.

Steve: Actually, if you were to walk up a step for every time you made a guess, you wouldn't need the Fitbit because you would be fit by the time you got the guess, yeah.

Leo: There you go. That's clever, yes. Just take the stairs.

Steve: Jason wrote: "Hi, Steve and Leo. Longtime listener and happy Club TWiT member."

Leo: Yay.

Steve: Thank you, Jason. He said: "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 some 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 would not be tagged with me by 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.'" And he put that in air quotes, so I don't think that's even his name.

Leo: We don't know his name. We don't know his birthday. We know nothing.

Steve: I loved that Jason put his own name in quotes, you know, 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, you know, for the sake of why not, you know, giving it a try. Anyway, he sort of prevented - he locked himself out from being able to do so.

An anonymous listener wanted to share some thoughts about leaving Windows. He said: "Hi, Steve. Please keep my name, company, and project private because it would be easy to reverse engineer who my company is." He said: "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" - and again, remember, really big company X, I know the name of the company, and it is really big. He said: "With respect to moving away from Windows to an open source solution: Much of my company's software, which is 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 will be going from a per compute device license to a per core license."

Leo: Oh, boy.

Steve: And I don't know about you, Leo, but I've got 20 cores.

Leo: Yeah. That's a massive increase.

Steve: He says: "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." He said: "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.

Leo: Wow.

Steve: So this person was actually, Leo, 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 kept hearing 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 more so than Microsoft.

And the message is, you know, the message was that they will be moving because Microsoft's policies appear to be predatory. "Predatory" was the word that several of our listeners independently used. And I thought, whoa. So, and I suppose it makes sense. If Microsoft can increase their profit and reduce the burden of support for all those pesky customers that they'd rather not have, then fine. Go to Linux. People are saying, okay.

Leo: Yeah.

Steve: TJ Asher said: "Steve, I heard Leo mention Jackpot Junction in that list of companies on the ransomware site."

Leo: Oh, yeah, yeah. They were one of the hacks or ransomware companies, yeah.

Steve: Right. He said: "That's a casino here in Minnesota. So I went to their website, and they have a big notice. It says: 'Slot machines and kiosks are currently unavailable. Bingo is canceled until further notice.'"

Leo: Oh, no.

Steve: "The special Bingo" - no, don't take my bingo.

Leo: No, no, not the bingo.

Steve: "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." Boy, this really hit them hard.

Leo: Oh, I feel bad for them.

Steve: "Full Deck is open for breakfast from 7:00 a.m. to 11:00 a.m., with regular menu from 11:00 a.m. until close. Table Games and Circle Bar will remain open. Thank you for your patience and understanding. We will provide updates as they are available."

Leo: They got hacked, all right.

Steve: And TJ signed off, saying: "Definitely looks like they got hacked. Keep up the awesome work. Regards, TJ." So for anyone who's interested, remember, I think it was, what was it, last week's podcast, I think it was GRC.sc/1019 was the shortcut that I created to take us over to Ransom List or whatever it was called. Oh, yeah, ransomlook.io. Yeah. GRC.sc/1019. And that's ransomlook.io. And, I mean, I looked again, and it's just - it's hopping over their recent posts on the left. It takes you to the listing.

Leo: Oh.

Steve: Yeah.

Leo: This is today.

Steve: Yup.

Leo: This is just today. These are all places that have been...

Steve: National Association for Stock Car Auto Racing, they're gone. Third Avenue Management gone. Crystal-D.com gone. Coop57 gone.

Leo: RoyalSaudiAirForce.gov.sa. Oh, wow.

Steve: Liberty Tax. They're going to be paying some tax.

Leo: Yeah.

Steve: CVTE.

Leo: This is the list you don't want to be on.

Steve: Oh, boy. And again, if any of our IT friends listening are having a problem with their CFOs, just say, okay, CFO, just go over here. Not one of these companies wants to be there, and they didn't give their CIO enough money.

Leo: Yeah. Yeah. Wow, incredible. And it was good to have that confirmation that, you know, we saw that casino on there.

Steve: Somebody listed there is SOL, yup.

Leo: I mean, not a good thing by any means.

Steve: No Bingo for Bongo.

Leo: No bingo for you.

Steve: No. Henrik Johnson said: "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's WAF (W-A-F), you know, Web Application Firewall, 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." So thank you, Henrik. So a better way to say it would be that Cloudflare is "fronting" for 20% of the Internet's website properties.

Harry Pilgrim said: "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."

Leo: Yeah, that's what I say. I never say "certificates."

Steve: Oh, okay, then it's I who am saying "certificate." But these are not "certificates."

Leo: No.

Steve: "A certificate is composed of uniquely identifying information," anyway, blah blah blah. He explains that. So thank you, Harry, for correcting us. I certainly stand corrected. But this gives me the opportunity to mention my absolute favorite SSH client and server solution for Windows-centric users, which is Bitvise, 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. 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 cannot recommend them more highly.

If all you need is an incredibly good SSH client for Windows, for accessing remote SSH servers, you can use theirs free of change. The Bitvise client is free. If you want a matching 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 the access to upgrades expires after a year. That server software will run forever. Mine's expired a few times, and they've had some updates, and I've thought, okay, I should re-up because I'm using their server very happily. I've been with them for seven years. I can attest that they are not constantly fixing mistakes.

Only very occasionally do they have something that they need to tweak. And normally it's for some edge case that doesn't affect me. But I want to stay current with them anyway. I could not be more pleased with them, and I cannot imagine ever having a need to switch. So just for the record, Bitvise, B-I-T-V-I-S-E, is my SSH solution for Windows.

Leo: That's one of the main reasons I'm not a Windows user is I need a command line that I can do things like that.

Steve: Ah.

Leo: I should say like this, and login to a remote server. I like having a command line.

Steve: I like it, too. It is, it's a good thing.

Leo: So I always, for a long time, I mean, I haven't used Windows in a while, but I used Cygwin. Is that, like, all done? Is that old hat? C-Y-G-W-I-N? Maybe it is. Bitvise looks pretty nice.

Steve: It's really nice. I mean, it manages our public and private keys, synthesizes keys. The server tells you it's never seen this key before. It tries multiple styles of authentication in sequence. You're able to maintain a list of previous SSH servers and select. It'll bring up a console window for you. So like when I SSH in to my FreeBSD Unix, I get a console window. Or when I SSH even into Windows, I get an admin prompt window. And I'm able to bring up a two-pane file copy so I can drag and drop files back and forth.

Leo: Oh, that's nice, yeah.

Steve: Anyway, it's just a great solution.

Leo: Bitvise. Free.

Steve: Highly recommended, Bitvise. David Spicer said: "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 safer."

Of course you're singing my tune; right? I've said, as long as you offer a fallback, then email continues to be the weakest link in the chain. I just logged into Hover a minute ago when you were giving our first advertiser, our first sponsor, because I wanted to see how much a .secure domain would cost. And I noted that right there, under my prompt for a one-time authentication, was "I don't have access to my authenticator." Well, okay. Then how good is this?

Anyway, he said: "My online banking site requires a one-time password code just to login once." He said, "I can view all of my account information normally. However, if I want to perform any money transfers, I am prompted for a new one-time code before I can do so. That made me think that this method might be useful with other online services that only support one-time password multifactor authentication login, such as Mailchimp.

"Even after you have signed in, if you wanted to perform a security relevant action, such as exporting data," which of course Troy got bit by, which is what made David think about this, "changing authentication methods, or viewing API keys, that would require a new one-time password 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's 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" - except today is extra good - "and your tools like SpinRite, ValiDrive, and the DNS Benchmark are amazingly useful. Really looking forward to buying the Pro version of the DNS Benchmark when it comes out for my lab environment. Have a great week. Thanks, David."

So I agree with David completely. Requiring the re-use of a one-time password or, you know, OTP token before proceeding with any extra-sensitive action after being logged in makes a ton of sense. And think about it. 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. Right? You know, why? We're obviously already logged in. In order for us to

even be presented with that opportunity of changing our password, we have to be logged in with our password. The site already knows who we are enough to allow us to be roaming around inside it. 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 most, you know, 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. And I think I know why. My presumption is that the reason for this is that most sites are still using some canned OAuth login authentication solution and have not bothered to build-in one-time password re-verification. Perhaps in time, you know, this will change, since re-prompting for one-time passwords I think makes so much sense. It really ought to be done. But his point's a good one. No one's doing it.

John Rostern said: "Steve, I've been a longtime Security Now! listener and have always appreciated your insightful commentary and analysis, mixed with some humor, on all things related to cybersecurity. I was a bit taken aback therefore by your somewhat dismissive comments regarding the Security Technical Implementation Guides (STIGs) in Episode 1018. The STIGs" - and they are at https://, and Leo, you should go there, public.cyber.mil/stigs - "represent an authoritative resource for secure systems deployment. The voluminous..."

Leo: Voluminous, yes.

Steve: Voluminous. There it is, voluminous. Thank you. I got started off on the wrong foot.

Leo: Yeah, you've got to start right, voluminous.

Steve: "The voluminous STIG documentation" - and it is voluminous - "and tools are provided free of charge" - in the upper right, click on STIGS - "free of charge including the Security Content Automation Protocol benchmarks. Misconfiguration has been and remains a 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."

Leo: Nice.

Steve: So thank you, John. I stand before you willingly chastened. I did not intend to be dismissive of the STIGs because I was not at all familiar with them. But I'm always wary, just sort of generally, 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. You know, value and effectiveness is never what I expect from government 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 the Security Technical Implementation Guides, I have a link to them, which John provided, in the show notes.

Michael Swanson said - and it appears that many of our listeners have encountered these STIGs. Michael said: "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 U.S. 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, et cetera), web browsers (Chrome, Firefox, et cetera), common applications (MS Office, Adobe, et cetera), 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 and so forth), 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."

So Mike, thank you. This makes absolute sense. I went over - oh. I know where I was, Leo. It was at STIGviewer.com (S-T-I-G-V-I-E-W-E-R dotcom /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 STIGviewer.com and then just choose "STIGS" in the upper - that's what I was thinking of, in the upper-right-corner top-of-screen menu - to see a huge alphabetically sorted list of very useful security-hardening checklists. I will be, my next Windows server will be, I think it's Windows Server 2022, which was the latest, the last of the Windows 10 equivalents. And they have a long list of things you absolutely positively want to do.

I already stumbled on one that was a little gotcha in IIS, some weird thing that was not blockable that would allow an undocumented protocol to get through. And I thought, whoa. And it worried me, like what else is in there? So I will definitely be going through the list before I deploy Windows Server 2022. It looks like a great resource. So thank you, listeners, for not letting me just blow that off because I didn't know any better.

Leo: Good.

Steve: And Leo?

Leo: Yes.

Steve: Let's not blow off our last supporter, sponsor. And then we're going to talk about, doo-to-doo, Multi-Perspective Issuance Corroboration.

Leo: Finally.

Steve: And why all Certificate Authorities gotta have it.

Leo: 1020 episodes, we finally got around to it.

Steve: Well, it didn't exist until last week, but okay.

Leo: Never mind. Our show today - well, in that case we're on it. We are on top of it.

Steve: Oh, baby. Oh, yeah.

Leo: Breaking news.

Steve: We got you some of that multi-perspective issuance corroboration. You betcha.

Leo: It's finally here. Steve, now, whatever the hell this is, multi-perspective issuance corroboration, it's time to dig into it.

Steve: That's right. Today's main topic was an outgrowth of an interesting change that the famous CA/Browser (CA/B), 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 certificates, and everything else that's relevant surrounding the increasingly crucial need for 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 - and this was a unanimous agreement - 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. Google wrote, because of course Google is an active participant in the CA/Browser Forum thanks to Chrome, and they have their own root program.

They said: "The Chrome Root Program led a work team of ecosystem participants, which culminated in a CA/Browser Forum Ballot to require adoption of MPIC" - which is the initials of today's podcast topic - "via Ballot SC-067. The ballot received unanimous support from organizations who participated in voting. Beginning March 15, 2025" - so that's last month, middle of last month - "CAs issuing publicly-trusted certificates must now rely on MPIC as part of their certificate issuance process." Whatever that is. "Some of these CAs are relying on the Open MPIC Project to ensure their implementations are robust and consistent with ecosystem expectations."

Okay. 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, and then we're going to digress. So Google said: "Before issuing a certificate to a website, a Certificate Authority 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 could 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 of Princeton University and others highlighted the risk of Border Gateway Protocol attacks and prefix-hijacking resulting in fraudulently issued certificates. This risk was not merely theoretical, as it was demonstrated that attackers did successfully exploit this vulnerability on numerous occasions, with just one of these attacks resulting in approximately $2 million of direct losses."

Okay. So "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 attacks."

Okay. So 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 how exactly 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 relied only 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, which is around 1500 bytes, into multiple individual packets. Each individual packet indicates where it's 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 big-iron Internet routers. 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. So that's the entire structure. That's it.

So upon a packet 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. There's too much incoming, all trying to go out of a narrow pipe outgoing. In that case, the router's incoming packet buffers would overflow, with nowhere left to temporarily store any 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 fundamental design of the 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 okay. Let's make it survivable. Let's design the protocols that place these individually potentially lost packets onto the Internet in such a way that a 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 that was sent out in a UDP packet, just sort of hopefully and blindly, if it's not received within a reasonable amount of time, the query will be retried and often reissued to all the other DNS servers that the client knows about. And this will continue, the retrying will continue until it finally gives up. But a lost packet will just simply be retried.

So, crazy as it might seem at first, every Internet protocol that generates and receives individual Internet packets assumes that its packets may 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. That's all it is. A whole bunch of routers, all connected to each other. 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 packet will deal with that.

Okay. So what does all 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, "it" the ISP has been assigned by the Internet's governing bodies and which its customers are busy using. BGP sends this information to all the routers that connect to the ISP's router so that they in turn know to forward any packets they receive on any of their other interfaces to the interface with which they connect to the ISP's router. After setting up their own routing tables appropriately, each of those routers in turn use BGP to forward their updated routing tables to all of the neighbors that they connect to, and so on and so on and so on, until eventually every big iron router anywhere on the Internet has received the information, the propagated 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 it works with astonishing reliability that we're all spoiled from now. 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 mistakes, innocent mistakes made with the Internet's big routers which, for example, for a few very hectic minutes might attempt to route all of the entire Internet's traffic through a bungalow in Myanmar. But, you know, perfection is understood to be impossible, so a system that's self-healing and resilient in the face of mistakes is what we want, and it's what we have today. And also through the years, the original vulnerabilities in these systems have been found, recognized, shored up, and improved.

So 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, my Certificate Authority, 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, and random gibberish data content, 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, with the proper contents, from the root of GRC.com. If that can be done, that proves to DigiCert that, whoever I am, I'm able to affect the content of the website located at GRC.com, which no one else is supposed to be able to 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 web server at GRC.com, it's just sending packets to, you know, DigiCert is 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 web server in California which has the IP address of GRC.com, and verifies the contents of a specific file which they created for the 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 web server 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 web server 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 address within that small network prefix. 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 my domain, GRC.com.

And here's the crucial point: The only way and reason this BGP router prefix-hijack attack works, which as Google mentioned has been shown to be real and effective and has proven to be a true problem, 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 web server 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. And now we know what that term means. With the researchers at Princeton University's Center for Information Technology Policy having demonstrated the real world feasibility of these BGP prefix-hijack attacks, all Certificate Authorities going forward must 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. CAs have a year to bring that number up to three, and from at least two distinct Regional Internet Registry regions. By June 15th of next year, 2026, that number grows to four, also from at least two RIR regions. And by the end of next year, December 2026, at least five remote network perspectives must be used in order to verify domain ownership and validation. 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 different locations that it's safe to say that this makes this mode of validation attack infeasible and takes it off the table.

Leo: Very cool.

Steve: So that is MPIC, Multi-Perspective Issuance Corroboration, you know, verifying ownership of a domain from multiple perspectives on the Internet in multiple locations.

Leo: You could still screw up the border router, though; right?

Steve: Yes. The Border Gateway Protocol, I mean, it's, you know, it's meant to be resilient, but it can happen.

Leo: Yeah, yeah. That's cool.

Steve: And I also wanted to note I heard your mention of the passing of the guy who...

Leo: Bufferbloat man, yeah.

Steve: And we talked about bufferbloat on the podcast and explained that it was messing things up because the Internet is designed to drop packets, and consumer router manufacturers thought, oh, we've got so much RAM, we'll have big buffers, and then it'll be great the packets aren't dropped. Well, it messed everything up.

Leo: That's not what you want.

Steve: You want to drop 'em.

Leo: Yes. He was only 59. He was a young guy. Let me see if I can pull up the story because we did, we talked about it on This Week in Tech on Sunday. And it was, it was a sad story.

Steve: Do we know how, I mean...

Leo: We don't know what happened, no. The only reason I knew what happened is Eric Raymond, ESR, posted something on X, eulogizing him. That's why I want to get the story because I've forgotten his name now. But that's kind of the story, in a way, is this technology that saved all of us, you know, bufferbloat was discovered and corrected, pretty much.

Steve: Yup.

Leo: By this one guy. So it's kind of a neat story. Let me see if I can - oh, shoot. Where is his name? I did so many stories. I'm looking through the show notes, and I don't see it. So, but yeah, it was a very...

Steve: Toward the end of a podcast.

Leo: Yeah, you'd think that these show notes would be in order, but - his last name was Taht, I think, T-A-H-T.

Steve: E, I think it had an E on the end?

Leo: Maybe it had an E on the end. Oh, now, this is going to make me mad because I do want - I do think we should bring it up real quickly.

Steve: How about we just - what if we google "bufferbloat"?

Leo: Google bufferbloat. Why is it not in the show rundown? That's the strangest thing. I must have accidentally deleted it after the show was over or something.

Steve: Okay. Wikipedia's got an entry, and I'll bet they give him credit.

Leo: Sure. Dave Taht (T-A-H-T) is his name. And here's the eulogy from Eric S. Raymond, who of course is a well-known open source guy, wrote "The Cathedral and the Bazaar." He says: "Dave Taht" - there's an umlaut over the "a" - "died yesterday, one of the unsung heroes of the Internet." He discovered bufferbloat and then went out and basically got router manufacturers to fix it. So it's less of an issue right now. So something to note.

Steve: Yeah. Wikipedia says it was initially described back in '85, and that of course predates this podcast. But it gained more widespread attention starting in 2009, and that's when you and I were together, and we said, hey, let's talk about this. It's cool.

Leo: Yeah. There's his X account. He lived in Half Moon Bay. There's not much more except that Eric Raymond message.

Steve: Lost him too young.

Leo: Yeah. And I guess he might have been on FLOSS Weekly back at March because Dave re-shared a FLOSS Weekly link. So, yeah. Unexpected, I think. I gather. Dave Taht, a guy whose name very few of us know, even those of us who know what bufferbloat is. But we do own him a debt of gratitude. So thank you, Dave.