Rowhammer Indelible Fingerprinting

Description: Could it be that yet another SQL injection flaw was found in the MOVEit Transfer system, and what more has been learned about last month's widespread attacks? What's a "Rug Pull"? What horrible conduct was the popular Avast AV found to be engaging in? Did China actually create their own OS? Version 1 is out! How many times can we say "TootRoot" while covering one story? What's the controversy surrounding the recent release of Firefox 115? Did Russia just successfully disconnect itself from the Internet? What are modern Internet honeypots discovering? How much of your life savings should you transfer into online cryptocurrency exchanges? (Okay, that's an easy one.) What did EU agencies just rule against Meta and Google? What happened to Apple's quickly withdrawn Rapid Security Response update? And after a bit of miscellany and listener feedback, we're going to look at the return of Rowhammering for the purpose of creating indelible fingerprints.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-930.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-930-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. There's a whole bunch of stuff to talk about. If you thought cookies were bad, wait'll you hear about Indelible Rowhammer Fingerprinting. Steve will also talk about his usage of Syncthing. He's got a pretty good workflow for backing up all of his assembly language. And then, yes, another critical SQL flaw in MOVEit file transfer. Wow. All of that and a lot more coming up next with Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 930, recorded Tuesday, July 11th, 2023: Rowhammer Indelible Fingerprinting.

It's time for Security Now!, the show where we get Steve Gibson in here to tell us what's going down. Steve's the host of the show for how many years now, Steve?

Steve Gibson: We're closing in on the end of year 18.

Leo: 18.

Steve: I think, you know, whenever I say that I get corrected by Elaine. She says, no, you started in - we started in '05, so when we finish...

Leo: 18 years, yeah.

Steve: Yeah. So when we finish 2023, and I think that's like next month, maybe, August? So that's good. We've almost reached maturity. We're almost adults.

Leo: Sorry, the Alexa was making noise. Okay. Yeah, you're going to be able to vote, but not drink.

Steve: Go figure. And defend our country's honor.

Leo: There you go. Why don't you stay where you are and do your job. You're doing a good job right where you are.

Steve: Yeah, I was, you know, a little nervous when we were - remember we had the lottery back when we were in high school.

Leo: I know.

Steve: It was like, uh...

Leo: I got a high number. What was your number, do you remember?

Steve: I was also a high number. I was in a lot of danger.

Leo: So a breath of relief; right.

Steve: And I figured, you know, Leo, they weren't going to put a rifle in our hands. They were going to say, oh, look.

Leo: Look, a geek.

Steve: Yeah. But we're going to make you wear a uniform, even if you're staring at a CRT. It's like, what? Why do I have this? What? Anyway...

Leo: I got a number of emails from people.

Steve: Yeah.

Leo: Who said Battle Dress Uniforms, or BDUs, are very comfortable.

Steve: That's good because, you know.

Leo: We were talking last week.

Steve: You know what, yeah.

Leo: About the guys in the Cyber Command wearing camo. You know, it would be better camo if it looked like the Matrix or something, you know, if it was green on black.

Steve: Ooh, yeah, yeah, that'd be good, yeah.

Leo: Yeah.

Steve: So for Security Now! Episode 930 - and by the way, Leo, we did confuse the world who are not in the U.S. about what happened to last Tuesday's podcast.

Leo: Oh, yeah, I apologize.

Steve: I also got a bunch of, like, where did it go? What what what what? But I have to tell you, I had a very nice week off, so.

Leo: Steve used to hate days off. Then he got married.

Steve: Yeah.

Leo: To a very attractive woman. And now he says "Give me some time off, I'll take it."

Steve: I'd be happy to stay home. So this is sort of an interesting evolution. Today's podcast is titled "Rowhammer Indelible Fingerprinting." And the title sort of says it all, but we're going to get into the details because that's where it really gets interesting. But before we get into that, we're going to ask the question, could it be that yet another SQL injection flaw was found in MOVEit's transfer system? And what more has been learned about last month's widespread attacks on all the users of that software?

What is a "rug pull"? What horrible conduct was the popular Avast AV found to be engaging in? Did China actually create their own OS? Version 1 is out. How many times can we say "TootRoot" while covering one story? What's the controversy surrounding the latest release of Firefox 115? Did Russia just successfully disconnect itself from the Internet? What are modern Internet honeypots discovering? How much of your life's savings should you transfer into online cryptocurrency exchanges? Okay, yeah.

Leo: I'm thinking a number - is there a number lower than zero?

Steve: Yeah, that is an easy one. What did EU agencies just rule against Meta and Google? What happened to Apple's quickly withdrawn Rapid Security Response update yesterday? And after a bit of miscellany and some listener feedback, we're going to look at the return of Rowhammering, this time not for a security breach, but for the purpose of creating indelible fingerprints.

Leo: Oh.

Steve: In other words, they can recognize your computer, and you can't escape.

Leo: Oh. That's not good.

Steve: Yeah, not good.

Leo: Never mind. I was all excited. Never mind.

Steve: And we do have a bizarre Picture of the Week.

Leo: Okay.

Steve: Which we will get to in a minute. Which I gave the caption "A Different Definition of Insanity."

Leo: I'm not looking until everybody gets to see it. I want to see it at the same time. All right, I'm ready, Steve. I'm going to take off the blindfold for the Picture of the Week. Oh, wait a minute, I have to push some buttons here before I can show that.

Steve: It'll take a little bit of visual parsing.

Leo: It's one of those things I have to look at for a little bit before I understand what I'm seeing?

Steve: Unh-unh, mostly because you won't believe your eyes.

Leo: Oh, come on. My little computer sometimes - there we go. Here we go. Okay. I am ready. I'm going to roll and scroll up. It's titled "A Different Definition of Insanity." What? Okay. Why don't you describe - I get it. It's so silly. I'm glad they made nice thick cables for this junction box.

Steve: So it's hard to - I think I understand what's going on because there's a little bit of a giveaway with this Post-it note down below the junction box. But so what we have is two massive conductor cables. I mean, like they look like they're an inch in diameter. They're huge. And they are running through, like, coming in from the left of the picture,

both of them in parallel, running through a junction box and then going out of the picture to the right. Then, and they're separated by maybe a couple inches. Then, okay. So like these things carry some ungodly amount of current.

Leo: These are big, yeah. There's one brown, one blue. Clearly, you know, this is 240 at least. It's a lot of voltage.

Steve: This is serious. So someone has removed the insulation from each of these, about two inches' worth of insulation, so we can see this massive copper cable that is being protected by the insulation. And then, apparently needing some light, they've wrapped some wire around, like one turn around each of them, and run the wires out the bottom of the junction box.

Leo: Oh, I didn't see the bottom. There's a little light bulb.

Steve: Where it's being used to power a light bulb.

Leo: A little one.

Steve: Yeah, like a regular, you know, Edison screw-in base light bulb. And so it's like, as if like this is the way you want to light the room or something, is like by opening up Godzilla's power supply.

Leo: By the way, the little wires aren't even soldered onto the copper. They're just wrapped around it and twisted.

Steve: It's just like, as I said, a different definition of insanity.

Leo: Now, what's this Post-it? Greater than underscore. What does that mean?

Steve: Yeah. I think that's meant to say greater than ground, as if to say, by the way, these wires are live.

Leo: Oh, they're hot. That's what the lamp is. It says these are hot.

Steve: Yes. I think this was all set up as some insane way of determining whether these are energized.

Leo: So don't touch them, kids. We've stripped off the insulation and opened the junction box, just to let you know, don't touch them.

Steve: Yes. Now, I'm sure all of us old school geeks learned a long time ago the trick of sticking your tongue on a nine-volt battery to see whether it was good or not. Do not do that here.

Leo: No, no.

Steve: Because you will have no tongue left.

Leo: Wow. Wow.

Steve: Anyway, this was just like crazy that this was done so informally. But the only way I can explain this is that this was some, like, poor man's way of determining whether the power was on on these Godzilla's revenge cables.

Leo: And presumably that box, because it isn't really a junction box, it's just got two wires going through it, it's built, I mean, it's for this purpose. Presumably they closed it up after they showed you their clever little hack. I hope they did. I hope they did because I don't know about you, but I'd be really tempted to lick it. I don't know why. I just want to; you know?

Steve: Oh, yeah. Anyway, thank you. We have such great listeners who are now on the constant prowl for gates in the middle of nowhere, bridges that go nowhere, and strange wiring. And, well, I have a few more, and I'm not going to give them away. But we've got some more fun things coming.

Okay. So anyone who understands the inherent danger of exposing the unnecessarily powerful SQL command stream to web servers, especially web server visitors, will not be surprised to learn that yet another instance of this extremely common and potentially devastating source of vulnerabilities has been uncovered in Progress Software's MOVEit Transfer system. Fortunately, the previous problems drew the attention of security researchers, in this case HackerOne and Trend Micro's Zero Day Initiative researchers. They decided to have their own look at MOVEit's code.

The good news is they responsibly disclosed their discoveries - still more discoveries - so that patches could be prepared and made available to Progress Software's beleaguered users. At the same time, Progress also patched another pair of high security vulnerabilities which were not SQL injection. There was one that allowed you to crash the MOVEit server. It's like, okay, fine.

So after literally an update per week - there have been three required updates about a week part, each which patched critical vulnerabilities - it's time to do it for the fourth time in four weeks. And I seriously don't know what I would advise MOVEit's users to do at this point. I guess I would feel extremely uncomfortable if I were to understand the inherent danger that this system presents to its users. As we covered three weeks ago in Episode 928, literally, I mean, truly thousands of Progress's customers were penetrated and did have their internal private data exfiltrated to Russia. Then they were extorted by Russia's Clop cybercriminal organization.

And the problem going forward is that due to the extremely poor fundamental security design of this MOVEit system, there's no reason to believe that the last serious data exfiltration problem has now finally been found. Sure, the system is doubtless far more

secure than it was four weeks ago, but the problem is this class of bugs, these SQL injection bugs, are slippery. And unfortunately bad guys are highly motivated to find a way in.

And as for MOVEit itself, we're back here on the topic of MOVEit hacks. The most current forensics conducted by Huntress Labs suggests that the attacks on those thousands of MOVEit-using companies appear to have been limited to just data theft and extortion. Which is a good thing because these were really powerful intrusions. The Clop cybercrime group has not yet been observed deploying ransomware into any of the incidents linked to its exploitation of MOVEit. Huntress also said that it has not observed the Clop gang expanding access into full network compromises, which, again, they could have done. The gang has apparently deliberately limited itself to infecting and infiltrating only the hacked MOVEit appliance itself. So as I said, things could have been far worse if a much more aggressive attacker had been found behind this.

And the other thing, too, that sort of gave us some pause is that the Clop gang from the beginning said, "We're not going to do anything with the data of governments or law enforcement." Almost as if they were a little put off by the aggressive response which some of the earlier ransomware attacks engendered. And they just kind of wanted to say, uh, well, we're bad, but we're not that bad. And so governments and law enforcement, if we get your data by mistake, we're just going to delete it, so don't come after us. No, and also educational institutions because there was a - it wasn't Carnegie-Mellon, it was some major university got exfiltrated. Anyway. So hopefully nothing came of all that.

I was scanning a news blurb about another crypto exchange mess. And it's like, yeah, yet another one. In fact, we have some numbers a little bit later. And everyone should understand that the bar has been raised fairly high for me by this point so that it's not all of these that I bother to bring up because it's just like, why monitor the flood? You know, you're wet, so you know that there's a water problem. But, you know, there's just so much of this going on. And it's safe to say that the entire crypto world is in pandemonium.

But then I ran across a term that's taken hold in the industry which left me shaking my head. And really there is some value to having everyone in our community of podcast listeners really appreciating the degree to what a mess things are. So here's the news that I encountered, which is interesting in itself, which concludes with this new bit of terminology.

Last Friday, July 7th, approximately $126 million worth of crypto-assets were mysteriously transferred from the accounts of cryptocurrency platform Multichain in what is believed to be a hack. This is according to blockchain security firms PeckShield, SlowMist - and we'll come back to SlowMist a little bit later - Lookonchain, and CertiK. And maybe you know you have a problem when there's four firms, like when monitoring blockchain security is immediately a business. That, you know, might be a bit of a problem.

So Multichain is sort of a meta platform which, as its name sounds, interconnects different blockchain platforms allowing users to exchange tokens across them. And as a result of this incident it was shut down, Multichain was shut down to investigate what happened. At the moment, neither Multichain nor any blockchain experts know exactly what it is that happened. CertiK believes some of the platform's private keys were probably compromised, although this theory has not yet been confirmed by anyone else.

Fortunately, due to quick intervention by platforms like Circle and Tether, $65 million of those $126 million which were moved from Multichain wallets were frozen pending the determination of what happened. So there was like, you know, if nothing else, as a

consequence of there having been so many problems, the industry is getting a bit better at just saying, okay, stop. You know, freeze everything. Let's figure out if this is good or not.

This is also the second time this year that Multichain has suspended its operations. It also halted trading at the end of May. At the time, the company said it couldn't perform server maintenance because its CEO had gone missing, and they didn't have access, as a consequence, to the entire platform. The rumor was that its CEO had been detained by Chinese authorities. So Multichain never indicated whether its CEO had rejoined the company. If the recent incident is confirmed, this would be the company's third hack in the past two years. It lost $3 million in January of 2022 and another $8 million in July of 2021.

Okay. So that's a bit of the background. Here's the way this piece of news concluded which made me just shake my head. It said: "Blockchain experts aren't ruling out a rug-pull either." And I thought, a what? A rug-pull? Yes. A rug-pull is when a cryptocurrency's platform developers run away with the money themselves. They "pull the rug" out from under their own operation. So what we have here is an industry that is so unstable that the term "rug-pull" has been coined as a shorthand to mean "the bank's vault got all filled up with value, and then the bankers just decided to take it." Because, you know, why not? Crazy.

And you know, Leo, just the other day we were both observing that neither one of us is generally a big fan of attorney-enriching class-action lawsuits. But I stumbled over some news that really makes you shake your head and maybe think, well, maybe there's a place for them.

Leo: I think, I mean, nobody makes any money; right? Except the attorneys. But it does call these companies to account, which can have some value; right?

Steve: Yes. And that is why I'm hoping that a big one lands on this firm. The Netherlands has a foundation known as the CUIC, which is for "Consumers United in Court." This CUIC Foundation has filed a class-action lawsuit against the quite well-known security firm Avast. CUIC alleges that, from its special position in its users' PCs - and not a few of them, either - Avast collected data on its users' activities, which it then sold to online advertisers without its users' knowledge or consent. Everyone who used Avast's AV products or browser extensions between May of 2015 and January of 2020 had their data collected and sold.

The list of products includes Avast Online Security, AVG Online Security, the Avast Secure Browser, the AVG Secure Browser, and the AVG Online Security extension. The collected user data was sold through a U.S.-based subsidiary named Jumpshot. Avast shut down Jumpshot back in January of 2020 after there was some hint of what was going on, and that was exposed.

So much as I generally despise class action lawsuits, in this case, as I said, I hope that the proverbial book gets thrown at them. This is a deep betrayal of trust. And it's sad. Avast has been around for 28 years, since 1995, near the beginning of the PC era. And it's currently in use by 435 million users. The only thing you can figure is that it must have been that the proceeds from collecting and selling data on the habits of that many individuals was just too much for them to resist. I wouldn't mind seeing them put out of business over this, frankly. Essentially, this was commercial spyware running in the PCs of those 435 million users.

Leo: It's kind of ironic. You get a secure browser. Oh, yeah, it's secure from everybody but us.

Steve: That's right.

Leo: We're going to keep an eye on you.

Steve: We're going to let you know if any other spyware comes in. But we're giving ourselves permission.

Leo: Avast acquired AVG. These are the two free antiviruses that a lot of people use because they were free.

Steve: Not quite so free, it turns out.

Leo: Well, it just reminds you, if you're not paying for it...

Steve: Uh-huh. Yeah.

Leo: There's some way they're monetizing this.

Steve: That's exactly right. Okay. So we've been observing that the era where it was feasible to actually create a new operating system from scratch has long since passed. The BeOS, you know, B-E OS, it occurs to me that that was probably the last entry that reached some level of maturity and had some adoption and critical mass. But of course then even it couldn't make it.

Leo: And that was the late '90s. I think people looked at that and said, oh, yeah, you can't write a new operating system now. It's just a Mac and Windows and Linux world, and that's that.

Steve: Right. And there have been some various small hobby efforts, but they never get very far, either.

Leo: Actually, BeOS is still around as Haiku. There's an open source project.

Steve: Yes, yes.

Leo: And of course Google wrote Fuchsia, but they're putting it on their IoT devices. It's not...

Steve: Well, and did they write it from scratch?

Leo: Yeah. Yeah, they said they did. No, no, they started from scratch.

Steve: Really.

Leo: It took them years, yeah.

Steve: Wow.

Leo: But it's not a general purpose OS.

Steve: Right. Well, and with Linux being open source, and with an incredible amount of effort already having gone into it, that's where any sane project would start today.

Leo: Start with a Linux kernel, absolutely.

Steve: And, sure enough, that's exactly the conclusion that China came to.

Leo: Yeah.

Steve: Recall that China announced that they were going to be replacing the West's Windows with "their own" operating system. It's actually Linux under its quite attractive covers. I have a link in the show notes, Leo, if you want to scroll through the pictures of the screen shots. I mean, it is, frankly, it's gorgeous-looking.

Leo: Yeah, a lot of people been saying nice things about this, you know.

Steve: It's called OpenKylin, K-Y-L-I-N.

Leo: It's Ubuntu; right?

Steve: Well, it's based on the Linux 6.1 kernel.

Leo: Okay.

Steve: They've been working on it for many years. It's available for x86, ARM, and RISC-V architectures.

Leo: They're clearly trying to make it look like Windows. It's even got a recycle bin here.

Steve: And rounded corners on the...

Leo: Yeah, yeah.

Steve: Yeah, I mean, it is really pretty. And so x86, ARM, and RISC-V. More than 200 companies, 74 special interest groups,. and 3,000 developers contributed to the effort. And the good news is, for any of our English speakers, it does have an English language option. So, you know, it looks very pretty. So anyway, so that answers the question about what China is doing, and it's their intention to say goodbye to Windows and hello to OpenKylin.

Leo: I'm all for that as long as they treat it as open. Russia did the same thing; right? Wasn't there a Red OS or something like that?

Steve: I don't know what they're doing. I did note that Firefox is the browser for this, so that's great, too.

Leo: Yeah.

Steve: We want to keep Firefox alive.

Leo: Yeah, that's right. Well, now you've got a billion new users.

Steve: Yeah. Okay. So I might not normally mention an extremely severe CVSS 9.9 vulnerability, except that it's in Mastodon. And when you learn that the vulnerability has been named "TootRoot"...

Leo: Oh, boy.

Steve: ...you really - you have no choice but to talk about it. So the Mastodon project has fixed the critical rootin' tootin' "TootRoot" vulnerability which and get this, it was bad allowed bad guys to commandeer any Mastodon server simply by posting a Mastodon toot containing a malicious multimedia file extension.

Leo: Oh, wow.

Steve: Uh-huh.

Leo: Oh, boy.

Steve: And as we know, that's not an easy thing to defend against, right, because the extension is, I mean, it's very difficult to get multimedia correct because multimedia codecs are massively complex interpreters. And interpreters are tricky. So the good news

is that once the trouble had been discovered, everything proceeded properly from there. Patches were released two days before the details were published. And in fact there was an announcement that there was going to be a release two days before that. So the whole system proceeded. And here's what's interesting, too, is TootRoot was discovered by security researchers at Cure53 because they were conducting a security audit at the behest of Mozilla.

Leo: Oh.

Steve: I don't know why Mozilla said...

Leo: Oh, Mozilla's launching in Mastodon. They're doing a Fediverse.

Steve: Nice.

Leo: That's why. That's why.

Steve: Nice. Then that would be it. Before we dive, before we jump in, we want you guys to give it a security audit. And sure enough, the audit came up with a potentially very serious vulnerability. And as we know, these guys were just looking at the Mastodon code. Bad guys could have, too. And so what the good guys can find, the bad guys can find.

Leo: Right.

Steve: Better than the good guys found it first.

Leo: And we're patched, if you're curious, twit.social.

Steve: Yeah.

Leo: And have been since the patch came out. We have a very good administrator.

Steve: Yeah. In total, Cure53 discovered four security flaws. And again, props to somebody for naming this TootRoot because, you know...

Leo: That's a good name.

Steve: That's a good name. Okay. So those of us who are using Firefox will find that we're now using Firefox 115. Since one of my two primary workstation machines, in fact it's the one that I'm sitting in front of right now, is still running Windows 7, and it's running perfectly. I also received the notice that this would be the final release of Firefox

for this machine, other than security updates. And that's fine since security updates is all I really need anyway. Edge and Chrome had both given up on me several months ago.

The somewhat controversial new feature in Firefox 115 is Mozilla's declared ability to remotely prevent arbitrary extensions to run on arbitrary websites.

Leo: Hmm.

Steve: Uh-huh. For some reason this has upset people, which makes no sense to me. I cannot imagine that Mozilla's motives would ever be anything less than pure. So if they know something I don't about some extension I'm running, and how some evil website I might mistakenly venture into is able to abuse that extension that I'm running, then by all means shut it down, and thank you very much.

Leo: That's what happens in the Chrome extension store. Everywhere; right?

Steve: Well, so there's blacklisting extensions.

Leo: Oh, it's not just in the store.

Steve: Correct.

Leo: Oh.

Steve: So this is the browser itself, and it's per website. So it's more fine-grained blacklisting. So in their explanation of this, Mozilla said: "Mozilla maintains an open ecosystem for add-ons, which gives developers many choices in how they create, use, and deploy their work. The same openness also provides malicious actors more opportunities, as well." And I think that's the key is they have some evidence that their openness is being abused by bad guys who are using the access to the code to find previously unseen problems, just like we were talking about with Mastodon.

Mozilla said: "While Mozilla can identify and block malicious add-ons discovered through tooling, reviews, and user reports, this is not always enough. Firefox version 115 introduced Quarantined Domains" - with capital Q, capital D, so that's their formal name for it, Quarantined Domains - " to protect user privacy and security when we discover significant security issues presented by malicious actors. This feature allows us to prevent attacks by malicious actors targeting specific domains where we have reason to believe there may be malicious add-ons we have not yet discovered. Users can also control this behavior for each add-on in the Add-on Manager (about:addons) starting with Firefox version 116." So the next major release. And they finished, saying, "We will be further improving the UI for users in future releases."

Okay. So 115 lacks some UI for this, but that's coming in 116, which I won't get on Windows 7, but I don't care because I'm happy to leave this in Mozilla's far more focused hands. Now, an example of an extension developer who drew some attention over like his upset with this is a guy named Jeff Johnson whose posting carried the headline "Firefox 115 can silently remotely disable my extension on any site." And it's like, yes,

that's by design. And why would they if it wasn't for the protection of the users of these extensions? Anyway, Jeff begins his grumbling by writing:

"Firefox version 115.0 was released on July 4th, but I'm not celebrating. I'm concerned about a new 'feature'" - he has that in air quotes - "in the release notes." And then he quotes them saying: "Certain Firefox users may come across a message in the extensions panel indicating that their add-ons are not allowed on the site currently open. We have introduced a new backend feature to only allow some extensions monitored by Mozilla to run on specific websites for various reasons, including security concerns." So that's what he quotes Microsoft saying. And he says: "For various reasons. That's quite uninformative and mysterious."

And he says: "I'm all in favor of giving users control over which extensions are allowed to load on which sites. Safari already has this feature on both macOS and iOS. My concern," he says, "is not about user control - little of which even exists in Firefox 115, as I'll show later - but rather about the remote control that Mozilla has now given itself." Again, to which I say "So what?" I mean, you know, that's good. That's what we want.

So anyway, Mozilla's in the driver's seat here. They must review and examine and digitally sign any extension before Firefox will even consider running it. And they also maintain, and Firefox enforces, a formal black list to disallow any previously signed extensions that are later found to be malicious. So all they're doing is adding an additional level of granularity and control by supporting the intersection of specific extensions and specific web domains. Sure. We may not understand exactly why. But they must have seen a need and decided that they're going to fill it for the safety of their users. So maybe the guy just needed to draw some attention to his site by posting this. I don't really understand.

Leo: Oh, no one ever does that. That's a...

Steve: That couldn't be a motivation.

Leo: Couldn't be the case, no.

Steve: How selfish. Okay. So as we've previously observed, Russia has been making noise about their own RuNet, which is what they call the portion of the Internet located within their territory. It appears that Russia now has a law which requires, and I get to say it, Roskomnadzor to perform an annual test disconnection of the Internet for the purpose of verifying RuNet's stability when it's functioning as a freestanding network. And they're claiming that happened in the very early morning last Wednesday, July 5th. However, people in Russia are doubtful.

Someone named Natalia Krapiva, she tweeted: "Last night, Russia tested disconnecting itself from the global Internet. On June 5th, around 2:00 to 4:00 a.m. Moscow time, authorities tested the Sovereign Internet system, which led to disruptions of various websites and government infrastructure services. Russian railroad services and food safety systems were reportedly disrupted after the Sovereign Internet testing on the morning of July 5th."

She then links to a Moscow Times article about this. However, someone named Oleg Shakirov quoted her tweet and replied: "Please don't take Russian or anyone's claims about testing disconnection from the global Internet at face value. Always remember about incentives within the system to exaggerate one's work. There are plenty of reports

that there were no universal break in connectivity. There is no real evidence that the drills caused disruption of Russian Railroads and the agriculture regulator. I mean this in a literal sense. There is no evidence cited in this Moscow Times piece. As I tweeted earlier, RZD" - and that's the Russian Railroad - "RZD problems started before the exercise and continue today, most likely caused by pro-Ukrainian DDoS-attacks."

And in fact earlier Oleg had tweeted: "The website and app of Russian Railroads have been disrupted for four days. Customers are told to purchase tickets offline. According to RZD, this is due to hacker attacks. The problem persists today despite yesterday's statement." He finishes: "The IT Army of UA implied its role."

And lastly, somebody else whose handle is @ug_sig tweeted: "Yes, a quick look at the sites that monitor Internet plumbing did not reveal any great disruptions during the time period identified in the reporting. One would have expected to see massive routing drops and other disconnections, but I didn't see any of that."

So yes, disconnecting from the Internet in any meaningful way for the purpose of isolating all of Russia from the West - have you seen how big Russia is? That would indeed be visible to anyone monitoring the Internet's operation. It seems likely that if anything was done at all, it was some token gesture, you know, like they pulled the plug on the Kremlin or something. You know, it's like, oh, look, no Russia. No Western connection. No Facebook. No evil Google; you know? And then they plugged it back in again, and that was their annual test. So, and in fact, there were reports of pro-Ukrainian forces taking credit for shutting down Russia's rail system using DDoS attacks. So again, I don't think that actually happened.

Leo: All right. Fully hydrated and ready to go with Act 2.

Steve: So Trustwave's SpiderLabs group carried out a six-month experiment creating a globally diverse honeypot network. They wrote: "To obtain a better perspective of attacks worldwide, Trustwave has implemented a network of honeypots located in multiple countries around the globe. By distributing honeypots in such a manner, we can gather a reliable set of information on the methods and techniques used by attackers and their botnets. In our pursuit to explore the current threat landscape, we established a honeypot sensor network across six countries: Russia, Ukraine, Poland, UK, China, and the United States."

Okay. So basically they established a widely geographically distributed set of listening posts across the Internet, collecting all of the incoming arriving packet traffic for a period of half a year. What did their six months of listening reveal? They found that fully 19% of traffic that probed their test honeypot network was malicious. So one in five packets incoming malicious. And of that malicious traffic, 95% came from IoT botnets which were out scouring the Internet trying to locate and exploit new devices and enlist them into their botnets.

So we have a world now where 19% of the traffic arriving at arbitrary IPs will be malicious. One in five packets is trying to do something bad. And 95% of those are from IoT botnets looking for a vulnerability in something that has been discovered trying to take it over. That's the reality of today's world.

Leo: Wow.

Steve: Yeah. It's not just the occasional packet; right? I mean, there's a lot of traffic coming to an active IP that is valid. And for 20% of that to be bogus, that's just astonishing.

So I know I've already talked about, just earlier in this podcast, the crazy losses being visited upon various cryptocurrency exchanges and services. But in preparing today's news I ran across a more comprehensive summary, sort of an aggregation of this, that I wanted to share. And remember that I had already mentioned SlowMist. A week ago, on Monday, July 3rd, three days after the first half of 2023 ended, the blockchain monitoring group SlowMist published their 2023 Mid-Year Blockchain Security and Anti-Money Laundering Report.

They started off by explaining: "This report delves into blockchain ecosystem security, summarizing key security incidents and funds recovery status in the first half of 2023. It aims to help readers identify suspicious transaction patterns and behaviors by analyzing typical cases, and explore the anti-money laundering landscape within the blockchain ecosystem."

Their report is lengthy and detailed, and there's no need to go into all of that. But here are some of the high points that will catch anybody's attention. Get this: More than $922 million worth of cryptocurrency assets, $922 million, so just shy of a billion dollars worth of cryptocurrency assets were stolen in the first half of this year, 2023, across a total of, and that is to say occurring during, 185 security incidents. 185 incidents. What is that? Is that one a day? That's one a day in half a year.

That dollar figure is, interestingly, less than half of what was lost during the first half of 2022. During the first half of last year, 2022, hackers stole $2 billion worth of crypto assets across, interestingly, 187 incidents. So the same number of incidents in both the first halves of this year and last year.

In total during 2022, that is, last year, Chainalysis is the name of the reporting firm, reported the loss of more than - so this is in 2022 - loss of more than $3.8 billion worth of assets. It's unclear why the first half of this year saw fewer funds, like half the funds, which were stolen compared to the first half of last year. But no one believes it's because cryptocurrency platforms have become more secure. As we saw, the total incident count was 185 versus 187. So essentially the same.

Nearly half the funds stolen this year, so far in the first half of this year, were taken from NFT, DeFi, and cross-chain bridge platforms, like that Multichain that we talked about before, that's like a new thing. And they lost a total of $487 million in 131 incidents. So about half of the total amount was just in that. And the year's largest hack so far was the Euler Finance incident, where the platform lost $197 million. That hacker eventually returned most of the stolen funds in one of the 10 incidents where attackers returned any stolen crypto. Usually that's in return for amnesty and a "bounty," which is typically a hefty percentage of their total take. So just saying.

It should be so abundantly clear that the world is still a long way away from figuring out how to do any of this securely. So it is really difficult to see how participating in any of this is worth the risk. It's just - it's just crazy. And I know none of our listeners are that nuts; you know? They're not going to stick their tongue across those two high-tension cables, either. But we all know somebody. Actually, we have some neighbors who were like telling us, you know, expounding on the benefits of cryptocurrency investment. It's like, and I just, you know, I bit my tongue, and not because it was sore from having stuck it on a 9V battery.

Leo: Did they ever try to sell you Amway or anything like that?

Steve: Oh...

Leo: That's the problem with the pyramid scheme. You've got to get others to join. Otherwise your initial investment is quickly worthless.

Steve: Yeah. So the world is continuing to struggle, well, actually I would argue the struggles are just beginning, over issues of cross-border Internet consumer privacy and the monetization of consumer data. Two examples. The European Union has just ruled that what Meta is doing is illegal under the EU's GDPR. The European Court of Justice has ruled in a case between Meta and Germany's Federal Cartel Office, concluding that Meta's interpretation of the EU's GDPR regulation is illegal.

The court sided with the German agency, which ruled in 2019 - as we know, these things always take a long time. Here we are, what, four years later - that Meta was bypassing GDPR privacy protections by taking data collected by various of its services without proper user consent and merging it behind the scenes. The aggregated data then allowed Meta to continue tracking German and EU users to feed its advertising business. The court's ruling bars Meta from engaging in such behavior in the future. So that's a first.

Leo: It also keeps Threads out of the EU, I think.

Steve: Uh-huh. Yes. Meanwhile, Sweden has just joined Austria, Denmark, France, and Italy in their growing crackdown on Google Analytics use. The Swedish data protection agency has fined two local companies - and these are hefty fines. I can't imagine they're going to get paid. The Swedish data protection agency just fined two local companies for their use of the Google Analytics service and has recommended against future use of the tool. A one million euro fine was handed out to Swedish telco Tele2, and a 25,000 euro fine to a local online retailer CDON. The fines were handed out because companies allowed Google Analytics to collect data on Swedish citizens which was then transferred to the U.S. As I noted, Sweden thus becomes the fifth EU member state to fine or recommend against the use of Google Analytics, which we talked about, you know, a couple months ago.

Now, assuming that the EU's legislation holds up, and they have every right to create and enforce whatever legislation they choose, it looks like it's going to become truly necessary for our large multinational Internet service providers to establish and run fully independent facilities which are able to demonstrate complete autonomy within each national region. And of course everyone's fighting against that, the Internet service providers, Facebook, Google, Apple and so forth, because it won't be cheap or easy to do so. But they're going to go kicking and screaming, but it does looks like the future. It doesn't seem like there's going to be a way for them to get around this because they really are aggregating user data and using it for their own benefit. The problem is right now it's crossing national lines.

Yesterday, and I didn't have a chance to listen to MacBreak, Leo, so I don't know if you guys talked about this, but Apple issued...

Leo: Oh, yeah.

Steve: ...and then almost immediately retracted an emergency update.

Leo: We did talk about it. One hour later they took it back.

Steve: Yeah, just after posting, and began pushing one of their new emergency, you know, it's the RSR, the Rapid Security Response updates. This one was iOS, iPadOS, macOS Ventura, and Safari. An hour later it was canceled and withdrawn. So what happened? According to reports posted to MacRumors forums, Facebook, Instagram, WhatsApp, Zoom, and other websites started producing warnings to their visitors about their updated Apple Safari browsers not being supported.

Okay. So the update repaired a WebKit zero-day vulnerability that had been discovered while it was under active exploitation. And that brings Apple's patched zero-days count to 10 for the year so far. Okay. But get a load of what happened. The previous versions of the various iOSes were 16.5.1 for the iOSes, and in the case of macOS Ventura, 13.4.1. What Apple did, since this was meant to just be a quickie RSR WebKit patch, was to leave the primary version number unchanged at 16.5.1 and to simply append a lowercase "a" enclosed in parentheses to the end of the version number.

So, as a consequence of this change, Safari dutifully began appending that same parenthetic "a" to the end of its User-Agent version string. And that was quickly revealed for some reason to deeply upset and confuse the User-Agent string parsers being used by a number of quite popular web servers.

So in the User-Agent string it said Version/16.5.2 is what I've got in my notes, with a "(a)" separated by a space. And apparently that was all it took. So Apple quickly retracted this update to minimize the damage that it was doing and advised people if they were seeing this problem to immediately roll back to the previous version. And they have said that they would be releasing a "(b)" update. And presumably, obviously it'll have the same zero-day fix; but they will, I guess, not put this into the User-Agent version string.

Leo: Not change the user-agent, yeah.

Steve: Yeah.

Leo: I kind of blame, I don't know, there's blame to go around because why are they being so picky about the user-agent?

Steve: I agree. I think that those web servers were using some common software that is common to them, which failed in parsing because...

Leo: There's an (a). What version is that? I don't know, no, no. Stay away.

Steve: Yeah. And, for example, I have an actual picture of the user-agent string which was captured by the updated software. And there's a bunch of stuff in parentheses.

Leo: Right.

Steve: AppleWebKit/605.1.15 (KHTML, like Gecko). So you would think that they're just going to ignore what's in the parentheses.

Leo: Yeah.

Steve: But no, apparently not.

Leo: Probably some, you know, complex grep that just fails; right?

Steve: Yup, it just stumbled; right.

Leo: Yeah, yeah.

Steve: So a couple of weeks ago we were talking about synchronized Synology NAS boxes and Syncthing and Sync.com and that free Windows app, @MAX SyncUp. Since I've changed what I'm doing, I wanted to update everyone, correct the record, and share a cool little bit of tech.

I soon recognized that the problem with using that @MAX SyncUp Windows solution was that, while it did provide bidirectional sync, it did not have an active agent running in the local Synology NAS. So it could not be proactively notified of any changes made there due to the inter-Synology synchronization, which I have between my sites, just as you have between your sites, Leo. And that turned out to be a problem.

Now, you mentioned, Leo, when we were talking about this that there was a native means for running Syncthing without needing some messy Docker or container encapsulation. So that sent me looking. And sure enough, I found it. As you said, by trusting an additional and very clearly trustworthy source of Synology add-ons, I found a native Synology build of Syncthing. So Syncthing is now running natively in each of my NAS locations.

Leo: Woohoo. Isn't that a great feeling? Oh, I love it.

Steve: Yeah, it is just right.

Leo: And you set it for, I hope, Receive Only. You can set a Syncthing folder to Receive Only, or Send and Receive, or Send Only. And I think that what you want to do on Synology is make it Receive Only, which means it will never delete anything. So it will only aggregate new versions. It will never delete old or, if you delete files on one machine, it won't delete them from the backup. That makes it a backup, a true backup instead of a sync.

Steve: Right. So that's sort of a different purpose than I have.

Leo: Oh, okay. You want it to sync sync.

Steve: Yes, yes.

Leo: Yeah, okay.

Steve: So I have an ASM tree on my machine in front of me, and an ASM tree on the machine in my other location.

Leo: Yeah.

Steve: And I want those two assembly language code trees to be kept synchronized.

Leo: Got it. You want to synchronize deletions as well as new files. Yeah, yeah, yeah.

Steve: Right, right, right. And so I've got Synology, I mean, I have Syncthing running on my various Windows machines. It's performing local-only LAN sync. And then I use the built-in, the inter-Synology NAS sync in order to keep those two Synology NASes the same.

Leo: Right. That's a really good solution. You don't need a cloud now at all, really.

Steve: No, no. And I, you know, I don't trust a cloud. Okay. So anyway, so I wanted to share how I'm handling the Internet exposure. Synology offers lots of "logon to your Synology NAS remotely" functions, and to which I say "thanks, but no thanks." I get it that that might be the right thing for some people, but it presents too great a vulnerability that can go far too wrong if the bits hit the fan. As we know, "featuritis is what bites us." So I have all of that "join your NAS to the happy Synology land community" stuff turned off. You know, my Synology NASes are little independent islands. What I want for my NASes is zero third-party and zero public Internet exposure, and it's possible to have that.

At each location, a pfSense firewall is what's facing the Internet. As I've noted before, pfSense is incredibly handy for performing static port mapping to bypass Cox's consumer port filtering. But in this case I'm using another nifty feature of pfSense, which is DNS-driven firewall rule updating. In other words, pfSense's firewall rules can track IP changes through DNS.

Each of the pfSense firewalls opens a single port which can only be connected to by the other endpoint, which is to say I have explicit IP to explicit IP openings through the pfSense firewalls. But that means that each endpoint needs to know the other endpoint's IP. pfSense also has a very capable and mature DynDNS client module. So I'm using a free DynDNS service to track the IPs of each endpoint for the other endpoint's use. As I noted, pfSense's firewall filters can be slaved to DNS lookups. So if either endpoint's IP should change, though that actually doesn't happen very often with a cable modem, that local pfSense instance will note that the IP it's received from the ISP, Cox, has been changed. So it will use its DynDNS client to update the DynDNS service with the news of its new IP. The other endpoint's firewall will then automatically update to allow a connection only from that new IP.

And yes, because I'm depending upon DNS, which is not fully secured, this solution is just ever so slightly less secure than the absolutely secure solution of manually configuring the IPs of each endpoint's firewalls. I could do that if it were absolutely necessary. But what this protects me from is the primary threat, which is having my use of Synology known in advance from an Internet-wide scan and then being vulnerable to a targeted attack due to any suddenly discovered vulnerability in Synology's software.

So anyway, I just wanted - it's cool that the pfSense can use DynDNS to publish its IP, and that it's also able to use DNS to automatically edit the filter rules in its firewall in order to track the IPs of anything that might be changing that needs to have access in. And that would also be very cool if you were traveling and wanted to have access into your network only from the IP where you currently were. So I wanted to share that little tip. And Leo, thank you for letting me know that I could run Syncthing on my Synologies. It is, you know, it is absolutely...

Leo: Yeah, I just think it's such a great program. And then, you know, I have it on every machine I use. So it's great.

Steve: Yup. Yup. And it keeps them all synchronized.

Leo: And I just, for people who want to use it for backup on a Synology NAS, when you set it up, you get all the folders. I basically said, you know, my Mac is the introducer, so just whatever it says it wants to share, accept. But then I go into settings in the folders, and I make them Receive Only. And according to Syncthing, that means it will still, if there's a change on the Mac, it will go to the Synology, and it will still synchronize that change to other computers. But if I have a deletion on any computer, it will not delete it on the Synology, which is what I want because it's backup; right? I want it to have everything ever, because the Synology is huge, and never - because my real fear with any synchronization tool is that you can synchronize deletes.

Steve: Right.

Leo: And if you delete a folder on one machine, and then it deletes it on all the other machines...

Steve: It propagates through the whole network.

Leo: Yeah. Then it's not a backup. So it's very important if you want to use the Syncthing as backup. And I do believe it works fine. It has for a long time for me. But if you set it to Receive Only on those folders on the Synology, then they'll be backup folders.

Steve: And I don't know if you've looked at it, but Syncthing also has a very nice and mature version tracking system.

Leo: I use the staggered file system, which is fantastic.

Steve: Yup.

Leo: Yeah, it's really good, yeah.

Steve: And, everybody, it's open source and free, so...

Leo: Free. It's the best.

Steve: And also widely multiplatform.

Leo: Yeah, yeah. Yeah, everywhere. I have it everywhere, yeah.

Steve: Okay, so a couple of little Closing the Loop pieces. George Balogi, he said: "Hi, Steve. Long-time listener of the show, and I owe a lot to you for all the IT knowledge I've gained. Thank you. Had a question and was wondering if you might know. Is it possible to read odd file system disks like those found in enterprise-level Xerox machines?"

Leo: Oh, boy.

Steve: He said: "Trying to prove or disprove a theory of what's left on these drives." And so, yes. It is definitely possible to look at and examine raw disk contents independent of any file system. Under Windows, my favorite piece of freeware is called HxD (capital H, lowercase x, capital D). The full name is HxD Freeware Hex Editor and Disk Editor. If you Google that, you'll find it. I have a link to it in the show notes. Made by a neat German guy.

If you can arrange to attach that drive to any Windows machine, HxD will allow you to "open" the drive without mounting any file system. And you then have all of your familiar Windows browsing tools, you know, a scroll bar so you can very rapidly scrub through the drive, a page at a time, search for strings and so on. And you would immediately be able to determine what was there, depending upon whether it's encrypted or not and so forth. But anyway, yes, George, absolutely possible. I would recommend HxD for Windows. And I think all of our various OSes have some means of allowing you to look at raw disk contents now.

Fairlane, posting as Skynet, he said: "I'm late watching last week's podcast, but thanks for sharing about the chirping of the mystery smoke detectors." He said: "Hilarious but not uncommon that you can't find it. I've had that happen at home. You said you didn't have any smoke detectors. What about carbon monoxide detectors? We have both, and when I can't find the one chirping, I will take out all the batteries of all of them and then start putting them back in one at a time until I find the one that starts beeping."

Leo: Oh, that's fun.

Steve: So I mention this because it generated a huge thread over in the newsgroup. It turns out that this is a very common problem.

Leo: Oh, yeah. And by the way, we don't know if it's a smoke detector. It could be a lot of things in your house.

Steve: It could. Actually, I know what it is.

Leo: Oh, you found it. Oh, good.

Steve: Yeah. It is a - for a while, well, actually several years ago the drain line on my air conditioning system clogged.

Leo: Oh.

Steve: And that caused the condensate from the inside coils to fill up, back up, and begin running water down into the people below me, and the people below them, because I am at the top of a three-stack.

Leo: The good news is, when you hear those screams from downstairs, you know who that is. There's no mystery about it.

Steve: So I decided, when this was all fixed, I added my own water leakage detector. And it's got a 9V battery in it. And were I to find it and stick my tongue on it, my tongue would tell me that it is no longer at 9V any longer. And so that's the thing that is chirping because it's trying to let me know that it's no longer reliably detecting water. However, the news, I don't need that information because I have since replaced the whole air conditioning system that has a built-in water monitoring and automatic shutoff system so that the water detector was superfluous. I took it out, and had lost it, and it is busy chirping forever.

Leo: Oh no. Oh no. Oh, jeez. So it's worse.

Steve: So I know what it is. I still can't find it.

Leo: It's under something.

Steve: Uh-huh.

Leo: But that's the problem is that frequency for some reason, it's hard to echolocate.

Steve: And if it only would like do it for a second, instead of like, what?

Leo: Yeah, yeah. And so you can get closer and closer. But I think because it's such, you know, normally high-pitched frequencies you can echolocate. But I think because it's so high-pitched it bounces off everything.

Steve: Yes. Wherever I go, it seems like it's somewhere else.

Leo: It's there. Yeah. Yeah, yeah, yeah. Oh, yeah, yeah, totally awful. Totally horrible. By the way, if you ever figure out, I replaced all my carbon monoxide detectors in the house the other day, they have americium in them, which is a radioactive material. And I've been talking with my friends. No one can figure out what do you do with this stuff? You don't want to put it in the landfill. Nobody takes americium. The best I could come up with is somebody said, if you have a friend who works at a hospital, they have hazardous radioactive disposal bins. Get them to put it in that.

Steve: Wow.

Leo: So get your feedback folks to help with that, too.

Steve: Okay. Oh, and we did have another person, Stephen McCalley, said: "Long-time SN listener. You probably already got the answer since I'm usually listening a day late, but the military requires wearing the 'Uniform of the Day' while on duty." He says: "For most branches that is the working Battle Dress Uniform (BDU)," he says, "which is camo." So yes, indeed.

Leo: Yeah. But again, the camo is for a jungle. They have desert camo if you're in the desert. They should have cyber camo.

Steve: It would be really good.

Leo: I want submissions for cyber camo.

Steve: We need cyber camo. And if we told Alex Lindsay, he'd have a photo of cyber camo by the time we were done.

Leo: Oh, yeah, he would. I'm sure. I'm going to work on it right now, come to think of it. I have Midjourney, too.

Steve: Good. So Simon Zerafa brings us our "Cringe of the Week." He said: "@SGgrc Attended a security conference that offers attendees the option to line-up for a help-yourself lunch. However, if that queue gets above a certain length, remaining delegates are provided an a la carte dining instead. This is to prevent Buffet Overflows."

Leo: Ohhh. Booo. That is a dad joke. Booo.

Steve: Yes, the old buffet overflow, when the line to get lunch gets too long.

Leo: That's terrible.

Steve: TWS said: "Hi, Steve. During a podcast from a couple of months ago, you mentioned a favorite Authenticator app that you use. Could you remind me what it was? Had been using Google Authenticator for many years, but I do not like these new changes." He didn't tell me what they were, but okay.

Leo: I don't like them either.

Steve: "Thanks for making a great podcast, and of course SpinRite. Looking forward to the new version." So it's OTP Auth. O-T-P space A-U-T-H. Since Apple's search is crappy, put something in, and what you're looking for is just a simple gray padlock. It's called OTP Auth. And it's just a gray padlock. And I love it. And again, it's also written by a neat German guy. I trust those Germans.

Leo: I will throw in my favorite, which is also open source and free: 2FAS. And it's iOS and Android. And as long as we're throwing things in, here are Steve Gibson and Leo Laporte in cyber camo BDUs, according to Midjourney.

Steve: My god, this thing works, doesn't it.

Leo: That was just one of several choices. Let's see what else we've got here. Here's a - I kind of like this one. It doesn't look like us. Which one do you like? The red? The green? It's definitely...

Steve: [Crosstalk] green.

Leo: Yeah.

Steve: I think that's...

Leo: Yeah, that's the one I upscaled. I thought that was very attractive. I don't know what the hood is, but that's your tempest-proof hood.

Steve: That's right, it's cold down here in Southern California.

Leo: Okay, sorry.

Steve: Wow.

Leo: Yeah.

Steve: Okay. So speaking of SpinRite.

Leo: Yes.

Steve: Sunday evening I posted the 33rd Alpha release and noted that it was the first release candidate.

Leo: Oh, baby.

Steve: Yeah.

Leo: We're getting there.

Steve: So this is the first release candidate for the DOS component of SpinRite 6.1. It appears to be finished. That release contained SpinRite's new embedded FAQ, and the people who read through it found three hyphen characters that were not displaying correctly. It showed as a lowercase "u" with an accent over it. But so far, everything appears to be holding.

This evening, while the newsgroup gang continues to pound on what we now have, I will begin the work of updating SpinRite's Windows app with the new USB formatting InitDisk technology that I started out creating three years ago, basically putting all the pieces together, and that will then move us to actual SpinRite 6.1, which once we know that it's working and tested, will just be replaced on the website, and that's what we'll begin offering. I'll let everybody here know that at that point that they're able to update their 6.0 with the easily downloadable and installable version. And Leo.

Leo: Yes?

Steve: Last Wednesday during Windows Weekly, something, I don't recall what now, caused you to note that for SpinRite's future I had chosen to...

Leo: Oh, yes. I was going to ask you about this.

Steve: ...finally leave DOS in favor of another operating system.

Leo: You had been using FreeDOS; right?

Steve: Yes, yes. Okay, so the first exposure many of us had to Intel processors was IBM's PC. As we know, it contained Intel's 8088 and then 8086 and so on. But as we also know, Motorola was bidding to have their 68000 processor chosen, and in fact that's what Apple used in their first Macintosh PC. To this day I still wish that's what IBM had

chosen since it was a truly lovely processor. I mean, it was just a gorgeous architecture, and I wanted to program it, you know, in assembly language. But that's not what we got. My point is that Intel didn't create their processors for the PC. The IBM PC project chose the already existing Intel processor family as the basis for their PC product line.

So that begs the question, what operating system did other early Intel x86 customers use who were not building PCs? You know, they were building elevators; trains, planes and automobiles; launching telecommunications satellites; and installing commercial HVAC systems. These things had no screens or keyboard. They were known as embedded processors, and they did their work mostly without any recognition of any kind.

Well, it turns out that in many cases those companies chose to use a real time operating system which had been created by - I don't know what it is with me and Germany these days, but a German named Peter Petersen. To give everyone an idea of who has always been using Peter's RTOS-32, just the beginning of his alphabetical customer list reads: "3M Company, Adaptec, Agilent Technologies, Airbus, Air Force Research Laboratory, Alcatel, AT&T Bell Lab, Audi, Bayer, Blaupunkt, BMW, Boeing, Bosch Telecom, Carl Zeiss, Carnegie Mellon, CERN, Daimler, Deutsche Telekom, Digital Research, Dow Chemicals, DuPont, Ericsson Mobile, ETH Zrich, Ford, Fuji Photo, Goodyear, Hewlett Packard, Honeywell Aerospace, IBM's Research Division, JPL, Lawrence Livermore National Laboratory, Leica," and on.

Leo: And that's just through the L's, kids.

Steve: And I skipped a whole bunch of like lesser known, yeah. Anyway, so you get the idea. And to that we can now add to that list Gibson Research Corporation.

Leo: So it's a real-time operating system.

Steve: Yes.

Leo: Okay.

Steve: Yes.

Leo: As opposed to DOS.

Steve: Exactly. This OS predates everything. Peter has been licensing it to the Who's Who of industrial Intel processor users around the world since the early '90s. And as Peter explained in his "going out of business" letter at the end of last year...

Leo: This is what cracks me up. I was accurate in that; right? Okay.

Steve: Right. The trouble was it is a finished product.

Leo: Right, like SpinRite. It doesn't need to get heavy.

Steve: Exactly. It perfectly interfaced his customers' provided code to their Intel-based hardware. And what's moreover, over the years he ran out of bugs because it was done. It was complete.

Leo: Wow. That's nice. Wow.

Steve: It was perfect. There was nothing left to fix. So his customers, faithful though they were through the decades, they stopped paying for annual maintenance because they never had any problems that needed fixing. So I purchased it lock, stock, and source. I own the result of those 30 years of embedded operating system software refinement.

Leo: Have you looked at the source code just to kind of...

Steve: Oh, it's just gorgeous.

Leo: It's in assembly, obviously; right?

Steve: No, it's all in C.

Leo: Oh, it's all in C, okay.

Steve: It's written in C. Which is fine because all of my code will still be in assembler. And it is, I mean, it is lean. 16K is the OS overhead.

Leo: Wow, wow, wow. That's amazing.

Steve: Yeah. So, I mean, it's exactly the right thing. It's able to boot on UEFI or BIOS. So that's really what I needed. And it runs, it is its own 32-bit protected mode operating system, and it has cloned a large subset of the original Win32 API, which Paul now pooh-poohs, but it's what I'm still writing to because it's like, you know, the real API. Anyway, so I cannot wait to get started on SpinRite 7.

Leo: Wow. And you don't care that it's out of support because you've got the source.

Steve: Yes, it's done.

Leo: It's done.

Steve: It's done. I got the source code. And, you know, eventually there will be some mass storage technology that follows NVME.

Leo: Right.

Steve: Because SpinRite 7 will be supporting NVME.

Leo: Okay.

Steve: And so, you know, for SpinRite 9 or 10 I'm going to want to be able to support that. So I have the source which will allow me to continue to evolve this operating system going forward.

Leo: Is it still - you still using 18, or is that available on it? Or is that - that's BIOS, isn't it, the interrupts.

Steve: Oh, yeah, yeah. BIOS is all gone.

Leo: So you don't have any BIOS dependencies anymore.

Steve: Correct.

Leo: And does it use RTOS to do the disk reads and writes? Or are you doing that all internally, low level?

Steve: I'm not exactly sure. It does provide a disk interface. And what's really nice is that it's got file system support.

Leo: Oh.

Steve: For FAT32, for all FATs and for NTFS. I'll need to add, you know, EXT and so forth for the others.

Leo: So normally SpinRite doesn't care about file systems; right?

Steve: Well, it doesn't today, but that's where it's headed.

Leo: Yeah, yeah.

Steve: You'll be able to tell it I want to recover this file that I can't read.

Leo: Ah. That's nice.

Steve: Oh, yeah.

Leo: Oh, I'm looking forward to that.

Steve: And why SpinRite the whole drive if only a quarter of it has files on it. Only SpinRite the files.

Leo: Yeah. Interesting.

Steve: So, you know, I've got lots of plans for 7 and beyond.

Leo: What do you use the OS for? I guess running the program; right? Loading and running?

Steve: Well, yes, exactly. It sets up the processor. It handles multiple cores. Basically it does everything that I don't want to bother doing. And like, you know, and you know me, I may end up replacing a bunch of it with assembler or fixing it or there. But it just sort of gives me a start. Also it's got vast support for network adapters, and it allows cross-network debugging. So I'll be able to run it on another PC, and it understands Visual Studio. So I get to run in my very nice Windows debugger mode while I'm actually operating code in the other machine. And because it runs over the network, if a customer has a problem, I'll be able to debug it on their machine no matter where they are. So there are some really very cool advantages going forward.

Leo: Nice. Nice.

Steve: We should take our final break.

Leo: Oh, yes.

Steve: And then we're going to talk about fingerprinting that we cannot get away from.

Leo: Kind of wish RTOS was our advertiser at this point. Love to do an ad for RTOS. So is Peter - what's he going to do? He's retiring?

Steve: He must be about my age because he's been at this as long as I've been at SpinRite. And, yeah, I mean, he wrote sort of a sad note. And he said, "Well, you know..."

Leo: That's it.

Steve: I think what happened is that COVID finally tipped over. I think companies looked at their bottom line and thought, you know, where can we trim the sails?

Leo: Right.

Steve: And it was like, well, let's just let our license expire because we don't need it anymore.

Leo: That's why I've lately been begging people to join Club TWiT. We don't want our license to expire.

Steve: Well, and the other thing, too, is that if you were starting today to build an HVAC or a CAT scanner or an IV drip, you would not put an Intel processor in it. You would put a little ARM chip, and that's what you'd use. So the problem is no new application is going to be using the old PC architecture. But I am. It's perfect for me.

Leo: There's a few legacies like you still around.

Steve: That's right.

Leo: That's nice.

Steve: That's right.

Leo: Well, I don't want you to retire either, Steve. So I'm...

Steve: No.

Leo: Now you've got me all excited about SpinRite 7.

Steve: It's going to be good. So at the end of February next year, during NDSS, which is the Network and Distributed System Security Symposium which will be held in San Diego, the recently completed work of a team of six UC Davis researchers will be presented. Their paper is titled "Centauri: Practical Rowhammer Fingerprinting." Another title might have been "The Creation of a DRAM Supercookie."

Here's how they described their findings in their paper's Abstract. They wrote: "Fingerprinters leverage the heterogeneity in hardware and software configurations to extract a device fingerprint. Fingerprinting countermeasures attempt to normalize these attributes such that they present a uniform fingerprint across different devices or present different fingerprints for the same device each time. We present Centauri, a Rowhammer fingerprinting approach that can build unique and stable fingerprints even across devices with homogeneous or normalized and obfuscated hardware and software configurations.

"To this end, Centauri leverages the process variation in the underlying manufacturing process that gives rise to unique distributions of Rowhammer-induced bit flips across different DRAM modules. Centauri's design and implementation is able to overcome memory allocation constraints without requiring root privileges. Our evaluation on a test bed of about 100 DRAM modules shows that Centauri achieves 99.91% fingerprinting accuracy. Centauri's fingerprints are also stable with daily experiments over a period of 10 days revealing no loss in fingerprinting accuracy. We show that Centauri is efficient, taking as little as 9.92 seconds, so less than 10 seconds, to extract a fingerprint. Centauri is the first practical Rowhammer fingerprinting approach that's able to extract unique and stable fingerprints efficiently and at scale."

Okay. So this is some brilliant work. And it's one of those discoveries that's immediately obvious in retrospect. Rowhammering is a subject that's come up over and over through the years for us. What we know about it is that, in order to obtain maximum storage densities from our system's dynamic RAM, the storage cells have been shrunk, and the number per unit area has grown. And effectively any margin for error has been deemed too costly and has thus been eliminated. What we wind up with is main system memory that can be pushed over the edge through deliberate abuse, where in this case "abuse" amounts to just hammering on one memory address which can, with distressing success, cause adjacent memory bits to spontaneously flip from a zero to a one or a one to a zero.

This breaks all of the rules since there's nothing inherently wrong with reading one memory address over and over. So through the years we've seen this unwelcome consequence of too much data being stored in too little space being very cleverly leveraged in many different ways to breach the protective hardware-enforced barriers isolating virtual machines from each other or imbuing unprivileged processes with full root kernel privilege.

And now today we have another consequence of this aggressive hardware engineering, the observation that tiny variations in the manufacturing of today's DRAM chips allow them to be uniquely identified in the field, thus enabling indelible fingerprinting. Where previous Rowhammer researchers needed to search physical DRAM to find a location whose bits could be flipped, these UC Davis researchers realized that identifying the exact location of such flipping, and of which bits were flipped in which direction, produced what could be used to uniquely and indelibly identify one specific piece of DRAM out of the multitude, and that would never change throughout the service life of that storage device.

So here's how they frame their accomplishment in somewhat greater detail. They said: "In this work, we investigate a stronger threat model where a fingerprinter aims to extract unique and stable fingerprints for devices with identical hardware and software configurations over extended periods of time. To this end, we aim to capture fundamental differences in the physical properties of the device's hardware as unique fingerprints.

"Our key insight is that a fingerprinter may be able to extract fingerprints from inherent differences that arise as a result of minute process variations in the hardware CMOS manufacturing process. As users seldom modify their device hardware, these fingerprints remain stable, as long as they account for differences resulting from process variation in the same hardware. While prior research has explored variations in internal clocks, GPUs and CPUs, we're the first to successfully leverage memory (DRAM) for fingerprinting."

You know, I was thinking about this. Remember back in the early days of hard drives where like the MFM drives came with a printout of where the defects had been located in the factory, and you were supposed to enter those defects into the low-level format so

that those sectors would be marked bad from the start. Well, back then no one was thinking about fingerprinting anything. But those defects were a fingerprint.

Leo: Sure. They're unique.

Steve: For the drive.

Leo: Yeah.

Steve: Yes. Because you were never going to have the same map on two different drives because they were physical surface blemishes that the manufacturing process was unable to, like, not to have any of. So this is the same thing. These are defects that are not bad enough to disqualify the DRAM from use, but they are bad enough to uniquely identify a specific DRAM chip from all others.

So they said: "We leverage Rowhammer to extract fingerprints by capturing the side-effects of process variation in memory modules. At a high level, 'hammering' a memory row - in other words, repeated read or write operations in a short time interval - results in bit flips in adjacent memory rows. In this paper, we show that the pattern of bit flips due to Rowhammer can be leveraged to build a fingerprint. We also show that the pattern of Rowhammer bit flips is sufficiently unique and stable to build a reliable fingerprint for the population of computing devices, billions of devices.

"To build intuition, we visualized the distribution of bit flips produced by executing Rowhammer at the same locations on two identical DRAM modules at two different points in time. And the results looked promising. The distribution of bit flips was reasonably similar on the same DRAM modules at different points in time while being noticeably different across the pair of modules. So that was our starting point.

"Centauri is a practical Rowhammer-based fingerprinting approach that exploits bit flip distributions to extract highly unique and stable fingerprints even among homogeneous devices with identical software and hardware configurations," meaning nothing else would allow them to be fingerprinted. They are essentially identical, but they're not using all the exact same DRAM chips. They're using their own DRAM chips over an extended period of time.

"Centauri overcomes three main challenges that make it then practical for fingerprinting: First, the bit flips triggered by Rowhammer are non-deterministic; in other words, hammering the same location does not flip the same set of bits. Thus a fingerprinter has to account for this non-determinism to extract stable fingerprints. We identify certain practical scenarios that exacerbate this non-determinism where comparing set similarity to match fingerprints falls short.

"With Centauri, we hammer the same locations multiple times to extract a probability distribution of bit flips as fingerprints. We then compare the divergence of these distributions that leads to better re-identification of devices, even where there is a drastic difference in the specific set of bits flipped." In other words, they deal with the fact that they're actually dealing with frequency distributions as opposed to specific bits flipped.

They said: "Second, fingerprinters are constrained by the abstractions provided by the operating system to allocate memory." Right. Like virtual memory, the memory that an app sees is not the actual underlying physical memory due to the page tables that exist between. So they say: "These abstractions provide limited access to contiguous physical

memory, and hide information about their allocation on the DRAM. Without root privileges, these constraints prevent fingerprinters from trivially tracking the location of bit flips to fingerprint devices.

"We use the insight from our measurement study that the distribution of bit flips in contiguous 2MB chunks of memory is unique and persistent to overcome this challenge. Armed with this insight, we sample enough 2MB chunks to guarantee access to the same chunk for fingerprinting." And that is, I think, quite clever, and a bit chilling since it means that this can actually be accomplished without privileges, like by a web browser.

"Third, memory modules implement mitigations against Rowhammer, such as Target Row Refresh," which we've talked about before. They said: "While prior research has demonstrated ways to craft hammering patterns to bypass TRR, they provide limited insights towards operationalizing them to trigger bit flips at scale. Centauri systematically identifies effective patterns for at-scale fingerprinting using Rowhammer.

"We then evaluate Centauri on a set of 98 DIMMs across six sets of identical DRAM modules across two major DRAM manufacturers. Centauri produces high entropy," meaning many significant bits of fingerprinting precision, which you want if you're going to identify a single device among billions. And, you know, if you've got 32 bits of entropy, well, we know that's 4.3 billion devices. And they said: "...high entropy, with a highest fingerprint accuracy of 99.91% corresponding to a precision across that population of 98 DIMMs of 100% precision, and recall of 97.06%. Centauri also demonstrates high stability with daily experiments to extract fingerprints from the same devices over a period of 10 days without any degradation in fingerprint accuracy. Our experiments show that Centauri only suffers a minor loss in accuracy of 0.9% in presence of external factors that are not under the control of fingerprinters" - it might be something like temperature variation, for example - "but affect the distribution of bit flips such as the CPU frequency," they said.

"We also investigate the trade-off between the accuracy of Centauri's fingerprints against the efficiency of Centauri's approach in terms of the time taken to extract fingerprints. Centauri is able to extract a fingerprint in as little as 9.92 seconds" - so just shy of 10 seconds, as I mentioned before - "reducing the overhead by more than 95.01% while degrading accuracy by just 0.64%."

And they finished: "Therefore, our key contributions include" - there are four things. "Practically extracting highly unique and stable fingerprints using Rowhammer. We practically demonstrate Centauri on the largest scale of DRAM modules in current literature. Second, handling non-deterministic bit flips. We handle non-deterministic bit flips by hammering the same memory chunks multiple times and using the divergence between probability distributions of bit flips to re-identify the same devices when they're seen a second time.

"Third, overcoming memory allocation constraints. We overcome memory allocation constraints by devising a novel sampling strategy" - that is, the 2MB chunks - "that guarantees access to the same chunk of memory for fingerprinting. And finally, operationalizing bypass techniques for Rowhammer mitigations. We bypass Rowhammer mitigations, like target row refresh, by identifying effective hammering patterns that can trigger bit flips at scale."

So their paper is 16 pages long, and I just shared an edited-down summary from about the first page and a half. Through the balance of their document they proceed to explain in absolutely complete detail how they pulled this off and how anyone else could, as well. There's no mystery left. Anyone who is sufficiently skilled and motivated who wanted to implement an indelible tracker for anything that uses DRAM, and I'll just note that's

everything, now has all the information required to duplicate this technology. So Rowhammer has struck again.

Leo: You know, in a way it's ironic because the original Rowhammer was kind of non-trivial to implement; right?

Steve: Right.

Leo: You had to be pretty sophisticated. This is easier to do.

Steve: Yeah, you're right, because Rowhammer depended on exact bit flips in order to get what you wanted to change.

Leo: Right.

Steve: Here, they just look at, like, a big spread of what got flipped and go, okay, just remember that.

Leo: So could you do this in JavaScript in a browser?

Steve: Yes.

Leo: Oy. Thanks for publishing that article.

Steve: Yeah.

Leo: Now everybody knows. Because they've been looking for indelible ways to fingerprint individuals for a long time. I mean, obviously cookies are no longer the way to do it. And this is why the cookie banner just pisses the hell out of me, because it's identifying something that's not a threat, when real threats like this exist.

Steve: Yes.

Leo: I mean, this is very straightforward, it sounds like. Oh, well.

Steve: Yeah. It's going to end up, you know. It'll be implemented in WebAssem because there you can do it at full speed, and it'll be a WebAssem module that websites load into your browser and run in order to see where you are.

Leo: Who are you? Oh, hi, Steve. Good to see you again.

Steve: Uh-huh. Uh-huh.

Leo: Yeah. You can block all the cookies you want, buddy. We don't care.

Steve: Yeah. And in fact when you think about it, we've been thinking in terms of fingerprinting browsers. But if you ran someone's app in your machine, like Instagram Desktop or some other app, the app could also - would obtain the same fingerprint.

Leo: The machine is unique, yeah.

Steve: Yes.

Leo: Not the browser.

Steve: The machine.

Leo: The machine is unique, yeah.

Steve: Exactly. So it's cross-browser and cross-app identification.

Leo: Nice.

Steve: Yeah.

Leo: Well, it's not that surprising. It's great work. I mean, it's amazing that they thought of this. And I love the analogy you used because now I completely understand what they're doing. You know, that makes sense.

Mr. Gibson, as always, chef's kiss. This show is a regular listen for everybody who wants to know what's happening in the world of computing, especially in the world of security.

Steve: Bye.