Is YOUR System Vulnerable to Rowhammer?

Description: An analysis of Telegram Messenger's crypto. A beautiful statement of the goal of modern crypto design. Who was behind Twitter's recent outage trouble? An embedded Firefox root certificate expired. Who was surprised? AI-generated GitHub repos, voice cloning, Patch Tuesday, and an Apple zero-day. The FBI warns of another novel attack vector that's seeing a lot of action. Google weighs in on the Age Verification controversy. In a vacuum, Kazakhstan comes up with their own solution. Was Google also served an order from the UK? Can they say? A serious PHP vulnerability you need to know you don't have. A bunch of great listener feedback, some sci-fi content reviews, and a new tool allows YOU to test YOUR PCs for their Rowhammer susceptibility.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-1017.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-1017-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Some really interesting topics. We've always wondered about the cryptography used in Telegram's Messenger. Well, now we know that what we thought is not very good. We'll also talk about did Ukraine really attack X.com? Why your Firefox might have said, hey, you've got to update us. And then we'll take a look at testing your PC for one of the worst flaws ever, Rowhammer, how you can do it as a way of kind of giving back. Plus we're going to get you some great listener feedback and some sci-fi recommendations, as well, from the great Steve Gibson, next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1017, recorded Tuesday, March 18th, 2025: Is Your System Vulnerable to Rowhammer?

It's time for Security Now!, yes, the show you wait all week for. Our man of the hour, Steve Gibson is here to fill us in on everything that's going on in the - what are you covering your mouth for?

Steve Gibson: So I don't talk over you while you're doing your intro.

Leo: You know, I don't - that, you know, talk over me all you want. People are not here for me. They're here for you, Mr. G.

Steve: So they're going to get a lot of that. We've got a, I think, a really interesting episode. Some researchers, I forgot where they are. Are they German? I don't know. We'll find out. It's a mystery right now. But there's three of them, I'm sure of that. And they decided...

Leo: Well, at least we know that much.

Steve: Oh, it's the Chaos computer group.

Leo: Oh, it is Germany, then, yeah, yeah, yeah.

Steve: They decided that no one had really done a large population study of the prevalence of Rowhammer. Rowhammer hasn't gone away. It's still dogging us. The idea being that, if you read, if you hammer on a given region of DRAM, you can upset the neighbors, which is true if you just hammer on your house, too. Well, you know...

Leo: Yes. Different kind of neighbor, but yes, yes.

Steve: Yeah, yeah. And so what we have now, it's over on GitHub, it's downloadable natively, you can install it on a USB thumb drive and run it and get a report on your specific system's susceptibility to Rowhammer attacks. And as part of this, you optionally upload anonymously your data to their cloud. You're able, if you don't like to do that, or if you want to look at what's being sent first, it writes it to the USB stick, and you're able to, you know, peruse it and go, oh, yeah, there's nothing here that I care about, and off it goes. You get a brownie point from them if you do that. It's a chance to win some lottery, but I think it's like two chicken sticks or something, I mean, it's nothing that you really care about.

But they're trying to encourage this because they would like to get a much larger sample size. What they realized was that while, yes, you can demonstrate this bit-flipping problem on random systems, we really don't know how big a problem it is. So anyway, everybody who's listening, and hopefully lots more who will find out about this, can run this test, submit their data, generate a much better sense for the prevalence of this. But that's not happening yet.

First we're going to talk about, oh, the long-needed and awaited and, oh, it's just poetic, Leo, analysis of Telegram Messenger's crypto.

Leo: Oh, only you would think it's poetic, but okay. It's a work of art, a thing of beauty.

Steve: Just we're going to have to pause and just steep in this for a while.

Leo: I thought we knew how they did it. I thought this was widely known.

Steve: We always knew it was crap.

Leo: Oh, Telegram. Oh.

Steve: Telegram.

Leo: Ah, yes. They rolled their own, didn't they.

Steve: They did.

Leo: They're not using NACL.

Steve: And it stinks.

Leo: Oh. Oh, boy.

Steve: Yeah. But the best thing of this whole part is the - and these are like a team of five crypto guys, several from ETH Zurich. And we've got a guy from Amazon, but he said, "I'm not affiliated with Amazon for this, I'm just a crypto guy." But they produced the most eloquent statement of why modern crypto is modern.

Leo: Yeah.

Steve: And it may even - you may get a little wet in the eyes, Leo.

Leo: Oh, teary?

Steve: It's really good. Also we're going to look at...

Leo: Crypt poetry.

Steve: ...the truth behind Twitter's recent outage trouble. There was a lot, I got a lot of feedback from our listeners about this expiring embedded Firefox root certificate. And the question is, who was surprised by that? Well, it turns out not so many people. Also we've got AI-generated GitHub repos, voice cloning, Patch Tuesday, and an Apple zero-day. The FBI has warned of another novel attack vector that's seeing a lot of sudden action, and it's one that had never occurred to me. So it's like, ooh, let's talk about this. Google has weighed in on age verification and all of that mess. And in a vacuum of age verification, of all people, Kazakhstan has decided to come up with their own solution. It's not wonderful. Also...

Leo: Isn't that where Borat's from? Maybe I - might have been wrong.

Steve: Yeah. I think it was his idea, in fact.

Leo: Yeah, probably.

Steve: Also, Google, was Google served with an order from the UK, as Apple was?

Leo: Ah. They wouldn't be able to say, would they.

Steve: That's what people want to know.

Leo: Yeah.

Steve: Can they say?

Leo: No.

Steve: Also we've got a serious PHP vulnerability that everybody needs to make sure that they don't have because...

Leo: Well, I don't have PHP, so I'm glad to say.

Steve: But, well, lots of servers have PHP on their backend, serving their pages.

Leo: That's true, yeah, yeah.

Steve: I mean, I do. The good news is...

Leo: Your forums; right? They're in PHP; aren't they?

Steve: Yeah, yeah, yeah. And but the good news is I wasn't vulnerable because of the way I set things up. But, for example, the default XAMPP stack is vulnerable.

Leo: Yikes.

Steve: And that's what lots of people use. So I've got to make sure you don't have that. I did take the trouble to update my PHP because the version I was running was vulnerable, but the way I was invoking it wasn't. So anyway, we've got a bunch of great listener feedback, some sci-fi content reviews, and then we're going to look at how you can find out about your own system's Rowhammer vulnerability. So, you know, it's just your average...

Leo: Just your everyday...

Steve: ...Security Now!. I come home after one of these, and I say to my wife, you know, I think maybe this one was a good one.

Leo: Every one is a good one, Steve.

Steve: She says, "Okay, honey, yeah."

Leo: And I might tell you my story, the story of hairpin NAT. Do you know what hairpin NAT is?

Steve: Oh, yeah.

Leo: Of course you do.

Steve: And in fact it's a way of solving the problem of not being able to access your IoT devices from an isolated network.

Leo: Well, it turns out I have a Comcast business account, that's what we use to stream, and they disable hairpin NAT in their router. And I, for the longest, for literally eight months now since we closed the studio, have been wondering why I can't get to my self-hosted Wiki by its name, only by its number. Well, I now know. They don't support hairpin NAT. Who would ever have thunk? Who would have thunk?

Steve: You know who does is the Ubiquiti routers.

Leo: Yes. So I'm using Ubiquiti behind the Comcast router. Comcast, because I have a static IP address, says no, you have to use our router. I might figure out a way around that because that's - they say, and actually this was going to be my question to you, we can save it, they say it's for security reasons they don't support it. I find that hard to believe.

Steve: No, it's for support reasons they don't support it. They don't want to try to explain to Martha or Jeffrey or whomever that, well, look, here's, you know, I mean, because it's tricky to understand that data goes out essentially on the other side of the router and then is able to do a quick U-turn and come back in as something else. It's like, what?

Leo: It's a good description. It is a hairpin. Just so you know, the symptom is I have - I'm running a server, a wiki server, internally, inside my network, on this Comcast router. It's using its static IP address because that's the best way to do it.

Steve: Yup.

Leo: But I can't reach it from here by name. DNS doesn't work. I can only reach it by number. But if I go outside, or after I turn off my - it works fine.

Steve: Works great.

Leo: Yeah. And I never heard of this. And so for the longest time I thought my server was broken. Anyway. You, of course - I should have asked you. Russell found it, our wonderful IT guy. He did a little digging. He said, "I think that they turned off hairpin NAT." Now, Steve, as always, I have sealed myself into a soundproof room before the show.

Steve: Thank you.

Leo: So that I cannot see the Picture of the Day. But I might - are you ready? Shall I roll up?

Steve: I need to tell you first that the caption that I gave this photo, this is one of those that will take a little minute or two to sort of absorb.

Leo: Okay. Yeah?

Steve: The caption is "The Nature of Legacy Technology."

Leo: Uh-oh. All right. I'm going to roll...

Steve: Like technology we're never able to quite get rid of, much as we might want to.

Leo: Yeah. Don't we know. This is Microsoft's sad song. Oh, my god. Oh, that is hysterical. Oh, my. Well, look at that, kids. Okay. You'd better tell people. That is legacy, boy, yes.

Steve: Isn't that wonderful?

Leo: Wow. There's nothing below it. Wow.

Steve: So once upon a time there was a phone pole. And it went from the ground up into the air.

Leo: Yeah.

Steve: As phone poles do.

Leo: They do.

Steve: And people began stringing wires.

Leo: Sure. As one does.

Steve: Isn't it wonderful? Oh, it's just wonderful. And so wherever this phone pole was located, it was a very busy region. And over time it accreted more and more wires, largely running north, south, east, and west, you know, sort of in the, you know, you can see them coming and going. And then something happened. We don't know what happened. But the phone pole, you know, lost its...

Leo: No longer necessary.

Steve: Lost its footing. Actually, you're exactly right, Leo. There were so many wires hooked to the top of this phone pole that some industrious person said, you know, I bet we really don't need the phone pole to go all the way to the ground anymore.

Leo: That's tensegrity. That's what that is, right there, in a nutshell. Holy cow.

Steve: So some brilliant person, or an accident, or we don't know what, but it was clear, very clearly cleanly sawed off below all of this transactional wiring happening at the top of the phone pole so that there's just no more pole below the phone.

Leo: Unbelievable.

Steve: Yeah. It's just wonderful. And the nature of legacy technology, you know, you can't get rid of it; right? I mean, you need it. But apparently they had to run a bypass or an underpass or something, right, or a pedestrian walkway.

Leo: You know, I'm looking at it. I don't know what's - they don't show what's below it. I'm just curious.

Steve: No, no.

Leo: But that's hysterical. And it obviously is working. It looks like all those wires have a nice, tight - they're taut. They're good.

Steve: Yeah, it didn't droop at all when they cut the pole out from under it. No, still there.

Leo: Wow.

Steve: That's just one of our goodies. That's a good one. Okay. So our listeners possessing long memories may recall how, well, repulsed I was by Telegram's design the

first time I looked at it. And we talked about it on this podcast. It was just a pile of made-up nonsense. I mean, it just didn't - it didn't obey any of the rules of cryptography. And since that was the general impression of it, which was shared by the informed crypto community - this was 11 years ago, back in 2014. Pavel Durov, who we talked about a lot back then, his response to the community's shunning of his solution was to say, okay, fine. You don't like what I just came up with in the kitchen table?

Leo: I think it was his brother who wrote it, as I remember.

Steve: I think, ah, I think you're right.

Leo: Yeah.

Steve: Yup. And so it was his fault. Pavel said, okay, fine. I'll put up a prize of $200,000 - and this was in 2014 when that was more money - to anyone who can decipher an encrypted message sent between two Telegram end users. You know, you don't like my crypto, fine. Here's 200 grand. Again, the crypto community was unimpressed because that was beside the point. It's, you know, it's about elegance. And it's about rule following, which is what you do if you want solid crypto, not someone dangling a carrot. So by 2014, and this was the point, we already knew how to solve these problems correctly, and Telegram wasn't it.

Okay. So for this reason I was very interested, and I knew our listeners would be, when I saw that a team of actual cryptographers had finally - and, boy, this was not easy. I think it's like 107 pages or something of crap that they had to wade through. Anyway, they took a good actual hard long look at what can best be described as the ad hoc cryptography which was invented out of whole cloth by Telegram. And I use the phrase "actual cryptographers" because the first thing that becomes clear to anyone looking at Telegram is that its designers were not.

Five cryptographers, one from King's College, two from ETH Zurich, one from Tel-Aviv University, and the fifth as I mentioned from Amazon, last Monday published a paper containing their findings which was just presented during the EUROCRYPT 2025 cryptography conference. I've got a link to the paper here in the show notes for anyone who doesn't mind scrolling because it is a tour de force.

Their paper's title was "Analysis of the Telegram Key Exchange," and its Abstract reads: "We describe, formally model, and prove the security of Telegram's key exchange protocols for client-server communications. To achieve this, we develop a suitable multi-stage key exchange security model along with pseudocode descriptions of the Telegram protocols that are based on analysis of Telegram's specifications and client source code. We carefully document how our descriptions differ from reality and justify our modeling choices. Our security proofs reduce the security of the protocols to that of their cryptographic building blocks." That's all proper, of course.

"But the subsequent analysis of those building blocks requires the introduction of a number of novel security assumptions, reflecting many design decisions made by Telegram that are suboptimal from the perspective of formal analysis." Which is a really nice way, a polite way of saying, you know, like, we did the best we could because we were just handed spaghetti.

Anyway, they continue: "Along the way, we provide a proof of the security for the variant of RSA-Optimal Asymmetric Encryption Padding used in Telegram, and identify a

hypothetical attack exploiting current Telegram server behavior," they said, "(which is not captured in our protocol descriptions)." They said: "Finally, we reflect on the broader lessons about protocol design that can be taken from our work." And that's where the poetry comes in.

Anyway, so 104 pages later - and remember, most of, like, the beautiful research stuff that we do here talk about share, I don't know, seven, 17 pages, not 104. I think there's 107. Anyway, this was not a short paper. They conclude under the poetic heading "The brittle monolith that is Telegram." But it's not just their heading that's poetic. Listen carefully here to how beautifully they describe the way cryptographic protocols should be designed, versus what they found lurking in the heart of Telegram.

On page 104 they conclude: "In theory, the design of a cryptographic protocol has the sole purpose of achieving the protocol's security goals efficiently. In actuality, however, to achieve this goal it must also achieve the goal of allowing at least a sufficiently motivated expert to convince themselves that the protocol achieves these goals." Oh, this is so pretty. "In other words, the central insight of what is commonly referred to as 'modern cryptography' is that a cryptographic design is also tasked with being easy to reason about. A fundamental paradigm of achieving this goal is modularity, where different components of the design can be reasoned about in isolation and then generically composed to establish overall security guarantees." Oh, just beautiful.

"This modularity is typically achieved by relying on building blocks that provide strong security guarantees on their own, as opposed to only and potentially in specific compositions, and by breaking the dependency between different components of a protocol by avoiding re-use of secret material." Okay, now, I'll interrupt here just to say that, obviously, reading between the lines, what they found was that just a bunch of goo was just kind of like thrown in a big pile and scrambled around and connected to itself. And it's like, here you go. I mean, and remember, that's what we saw back then.

Anyway, they said: "Telegram's failure to achieve this design goal is the root cause for the limitations and complexity of our proofs and our seeming need to reach for unstudied assumptions on cryptographic building blocks than would otherwise be necessary. We will now discuss these issues and highlight several of the main Telegram design choices and their effect on our proofs of security. We begin with mere complications, then move on to limitations and seemingly necessary ad hoc assumptions. We finish by briefly recapping our hypothetical attack. We also discuss" - this is after 104 pages of getting up, leading up to this. "We also discuss design choices that led to these issues and note that the same design choice often led to several different difficulties for arguing for the security of Telegram, leading to necessary repetitions in what follows."

In other words, they're trying to do the best they can when given a mess. And we're, like, trying, like trying to agree that this thing was secure, but it wasn't easy. And several pages after that, under the heading "Reliance on Unstudied Assumptions," they added: "In Appendix C we describe several unstudied ad hoc and new assumptions that we used in our proofs. These assumptions could have been avoided if, for example, collision-resistant hash functions like SHA-256 or SHA-3 had been used instead of SHA-1" - meaning that's what Telegram is using, meaning it's not collision-resistant today - "and if proper key derivation functions had been used." Meaning it doesn't. So in other words, the cryptographic design of Telegram is a mess at a time when "a mess" can, and for very good reasons should, be avoided.

Telegram is likely secure enough for everything and everyone who's using it and relying on it. No one is saying it isn't. But its design actively fights against that actually ever being proven. So I suspect that Pavel's $200,000 reward, at least for the foreseeable future, is secure, as is Telegram. But there was no reason to just do it this way because, by the time they were designing crypto, it was already well established how to solve all

these problems, and they just didn't, you know, Pavel's brother, as you remind us, Leo, just said, eh, you know, I'm just going to - we're going to do our own thing because, you know, who will ever be able to prove it isn't. And he's right. No one can.

Those of us who watched the early rise of Twitter will recall the frequently seen "Fail Whale." Its appearance usually indicated that the service, which was struggling to grow fast enough to keep up with its exploding demand back in the early days, was temporarily unable to do so. That is, I mean, there was just too much desire for it. But the good news is those days are now long past.

However, last week Twitter was on the receiving end - and someone wrote back and said, Steve, why are you still calling it Twitter? Well, because I started with a retrospective, I suppose. But I just - the only problem I have with X is that it's so unspecific. I mean, for what it's worth, the tech press is still saying Twitter. And when you say "X," you're almost compelled to say, you know, "that service that was formerly known as Twitter," as if you're talking about Prince, that's now a strange glyph.

Anyway, Twitter was on the receiving end of a widespread high-bandwidth DDoS attack. And as we know, widely sourced, very high-bandwidth attacks are what's now required to take major sites and services down. In the case of last week's attacks, those who track such things, and there are a bunch of different groups who do, saw massive traffic originating from IP addresses in the United States, in Vietnam, and Brazil, as the top three among many other countries. So I was annoyed when Elon Musk later told Larry Kudlow, during an interview on Fox Business Network, that the attack came from Ukrainian IP addresses.

What actually happened was that a group which offers DDoS attacks for hire, named Dark Storm Team, took credit for X's Monday outages. I don't have any problem when someone has a differing opinion. But Elon could have either said nothing, or said he didn't know where the attack originated or why it was launched. You know, it would have been even better, and accurate, to say that, like most modern attacks, they come from all over the globe. And I get it that he's very busy. And I would imagine he probably didn't have any actual information at all. And he shouldn't be expected to know everything. Like I said, he's busy. But singling out and naming Ukraine as the source of the attack, first of all, was not true, at least from a bandwidth standpoint, which is knowable. And of course doing so appears to serve a current political agenda.

Leo: Yeah. It was propaganda. He, probably knowingly, lied. I don't - I can't understand how he could not know that it's not true.

Steve: I just think he's busy. I mean, you know...

Leo: Well, he's busy.

Steve: Larry said, hey, Twitter was down. What about that? And he should have said "I haven't been brought up to speed yet." I don't know. Anyway, for what it's worth, we do know that it wasn't IP addresses in Ukraine. So I just wanted to clear that up.

Leo: In fact, there really weren't many coming out of Ukraine.

Steve: No.

Leo: Ukrainians don't have much Internet.

Steve: No, exactly.

Leo: It's not where you would go if you wanted to do a DDoS attack.

Steve: No. And frankly, I don't think you can DDoS anyone through Starlink because it doesn't have that much bandwidth.

Leo: Right.

Steve: You need landlines that get warm with all the packets that are moving through them. So interestingly, last Friday a critical Firefox root certificate expired. Earlier last week, and this is what generated so much feedback from our listeners because everyone knows I'm a Firefox fanboy, Mozilla wrote: "On March 14th, 2025, a root certificate used to verify signed content and various add-ons for various Mozilla projects, including Firefox, will expire. Without updating to Firefox version 128 or higher, or the ESR, you know, the Extended Service Release, 115.13 or later for ESR users, including Windows 7/8/8.1 and macOS 10.12-10.14 users, this expiration, that is, the expiration of this root cert, may cause significant issues with add-ons, content signing, and DRM-protected media playback."

Now, just to be clear, this is a root certificate, not the way we normally think of it, not like a public root. This was a private root embedded in the Firefox EXE. So that's why it was necessary to have an up-to-date version of Firefox. Mozilla said: "If you don't update, Firefox features that rely on remote updates will stop working, and your installed add-ons will be disabled. DRM-protected content, such as streaming services, may stop playing due to failed updates. Additionally, systems dependent on content verification could stop functioning properly." In other words, lots of bad stuff.

They said: "This update is necessary for all Firefox users running versions earlier than, as I said, 128 (or ESR 115.13), including those using Firefox for Desktop on Windows, macOS, and Linux, as well as Firefox for Android. If you were sent to this article through an in-app message in Firefox, it means your browser version is outdated and needs to be updated."

Okay, now, since I'm still using, actually I'm sitting in front of it right now, Firefox on a Windows 7 machine, I was initially concerned. But I just checked, and my ESR edition had already updated itself well past that point. It's currently at 115.21.0esr. And in researching this further, it became clear that, unlike those sites which, you know, we sometimes see, I won't say often, where their TLS certificate expirations clearly are catching them by surprise because their site suddenly went offline and it's, like, whoops, we fired the guy that normally updates that every year, in this case Mozilla was not taken by surprise by this.

The mainstream v128 edition, which was recent enough, and that ESR release which Mozilla said would be needed, that 115.13, were both first made available on July 9th of last year, 2024. So, like, nine months ago; you know? Anyone who hasn't updated their Firefox even once since then would have no one to blame other than themselves if something were to go wonky with their client. What this meant was that Mozilla was just reminding everyone for the sake of doing so, a few days before that certificate's

expiration, which was formally retired nine months before, nine months ago, that if for any reason somebody might still be running a Firefox, you know, from last summer, then various important things might stop working.

Leo: You know, this could happen to me, though, because Firefox is not my primary browser anymore, but I have it on my machine.

Steve: Yeah.

Leo: If you never launch it, it never gets updated; right? So it's not inconceivable that you could, you know, have it sit there for a year.

Steve: I think, actually, I think it updates at launch.

Leo: That's the thing. It would update as soon as I launched it; right?

Steve: Right.

Leo: Or does it say, hey, restart to update, because I see that on Chrome.

Steve: We don't get that with Firefox.

Leo: Okay.

Steve: Unless you go to the About box. But normally it says, you know, you're updated. I think that where someone would get caught out would be if they had some version of Firefox, or, I mean, some running instance that was never restarted, like someone actually sent me a picture of a Firefox error message on a, like, Wendy's fast food drive-through kiosk. And it was, you know, like Firefox was unhappy about something. But so, you know, there might be an instance where it would just have been running for months on end and never restarted.

Leo: A kiosk would be exactly that; right?

Steve: Right.

Leo: Yeah, yeah. I have to show you, Steve. Somebody in our Club TWiT just showed us he's watching Security Now! in the barbershop.

Steve: Ah.

Leo: Put away the Playboys, guys. We've got Steve Gibson. Isn't that awesome?

Steve: Wow. That's some crazy barbershop.

Leo: Isn't that awesome? That is - this is a Club TWiT member who I think is the barber. His name is Sirio Barber.

Steve: Okay, well, that would explain it, then.

Leo: So I think it's his shop. Anyway, thank you, Sirio Barber.

Steve: Because really, you know, for most people getting their hair cut, if you can fall asleep during that, that's good. You know.

Leo: I get sleepy anyway getting a haircut. No, this would keep you awake, Steve, keep you awake.

Steve: Uh-huh.

Leo: Uh-huh, uh-huh. All right. On we go.

Steve: So we knew it was going to happen. And it's also probably little surprise that it happened not long after AI became the big buzzword. An unknown threat actor has deployed a large number of malicious GitHub repositories which infect users with malware. That's not such news. Trend Micro says descriptions for the repositories have been generated using AI tools. So we're beginning to accelerate the rate at which bogus GitHub malware repos are created and descriptions are created, hoping to catch unwitting people looking for solutions.

The malicious repositories infect users with the SmokeLoader, which then deploys the Lumma Stealer malware to exfiltrate users' credentials because they're looking to get developers' credentials in order to launch supply chain attacks to infect their own actual valid repos and get their stuff widely distributed. So beware of repos that actually, you know, they don't look like they're written by some Russian national trying to write English anymore.

Leo: Oh, no, they're good now. They're grammatically perfect.

Steve: Oh, boy.

Leo: And they sound that way, too.

Steve: A Consumer Reports study found that Speechify, Lovo, PlayHT, and Descript made no efforts to ensure that users had consent to reproduce another person's voice. So those are four out of the top six voice cloning apps don't have any problem if you reproduce someone's voice without their permission. They are, as I said, they are the top

four out of - those four out of the top six have no protections against abuse. They allow threat actors to easily clone anyone's voice, you know, given a sample. Consumer Reports study also found that voice cloning scams are seeing a wider adoption across the fraud landscape. You know, where it sounds like your Grama is calling and asking for some money.

Leo: You know, it's so funny because my mom's stock brokerage, I won't say the name, keeps pushing me to use voice identification.

Steve: It is so yesterday. I mean, it was just - it's a bad idea.

Leo: Yeah.

Steve: Wow. Like, I mean, first of all, it was never good; right?

Leo: Right. That's my thought. It's convenient, I guess, but yeah. No.

Steve: Maybe it just puts people off, like, oh, you know, some Russian is trying to scam you.

Leo: Oh, they use voice identification.

Steve: Then it's like, okay, I guess I won't, I'll go somewhere else.

Leo: No.

Steve: Last Tuesday Microsoft patched a modest 58 vulnerabilities, among which six were actively exploited zero-days. You know, that's only a third of what they've done recently, Leo, so that's like, yeah, okay. We'll wake up. These was a Windows Win32 Kernel Subsystem Elevation of Privilege vulnerability, Windows NTFS Information Disclosure vulnerability, the Fast FAT File System Driver Remote Code Execution vulnerability, NTFS Information Disclosure, another one of those, and an NTFS Remote Code Execution vulnerability, and Microsoft Management Console Security Feature Bypass. So those were all being exploited as zero-days among 52 others. So, you know, update when you can.

Apple also patched a zero-day in their WebKit, affecting both iOS and macOS. And Apple did describe it as an extremely sophisticated attack, so not easy to do. But, you know, they fixed it.

Now, this bit of news was interesting to me, had never occurred to me. The FBI is warning that their agents are increasingly seeing scams involving free online document converter tools, and they posted a note saying that "We want to encourage victims to report instances of this scam." They said: "In this scenario, criminals use free online document conversion tools to load malware onto victims' computers, leading to incidents including ransomware."

FBI Denver Special Agent in Charge - I wonder, Leo do they have any non-special agents? Or are all their agents special?

Leo: I should think they are all special agents, come to think of it.

Steve: I think they're all special agents.

Leo: I think they are.

Steve: You wouldn't want to be like not the special agent.

Leo: You don't want to get that one.

Steve: You may not always be special agent in charge, but you could be special agent.

Leo: Right.

Steve: I think they're all special. Anyway, this guy's name is Mark Michalek, and he said: "The best way to thwart these fraudsters is to educate people so they don't fall victim in the first place." Amen to that. "If you or someone you know has been affected by this scheme, we encourage you to make a report and take actions to protect your assets. Every day we are working to hold these scammers accountable and provide victims with the resources they need."

So the FBI said: "To conduct this scheme, cybercriminals across the globe are using any type of free document converter or downloader tool. This might be a website claiming to convert one type of file to another, such as a .doc into a .pdf. It might also claim to combine files, such as joining multiple JPG files into one multipage PDF. The suspect program might claim to be an MP3 or MP4 downloading tool."

They said: "These converters and downloading tools will do the task advertised, but the resulting file can contain hidden malware giving criminals access to the victim's computer. The tools can also scrape the submitted files for personal identifying information, such as" - I don't know who would have a Social Security number in such a file, but okay - "dates of birth, phone numbers, et cetera. Banking information, cryptocurrency information (seed phrases, wallet addresses and so forth), email addresses and passwords."

And they finish, saying: "Unfortunately, many victims don't realize they have been infected by malware until it's too late, and their computer is infected with ransomware, or their identity has been stolen. The FBI Denver Field Office encourages victims or attempted victims of this type of scheme to report it to the FBI Internet Crime Complaint Center at www.ic3.gov."

Leo: By the way, I did a search. Not all FBI agents are special agents.

Steve: Oh.

Leo: Special agents are the criminal investigators or detectives who, in other words, you might have the tea lady is just an agent, not a special agent.

Steve: So would you say, like, FBI generic agent?

Leo: Yeah, there are agents.

Steve: Oh.

Leo: Other employees of the FBI who handle administrative tasks, paperwork, or phone calls may be broadly referred to as "agents," but are not "special agents."

Steve: So I guess everybody is an agent. That's what you are. You're not an employee. You're an agent.

Leo: Well, I wouldn't go that far, either.

Steve: You think there are non-agent employees?

Leo: You can't be arrested by anybody but a special agent.

Steve: Ah.

Leo: They're senior to the agents.

Steve: Got it.

Leo: But there may also be other jobs. I'm sure the person who empties the trash in the offices is not an agent.

Steve: That's a good point.

Leo: I would think. I don't know. I just - I asked AI. AI told me that.

Steve: That's good. Well, we're going to believe it until we learn otherwise.

Leo: Until we learn - till it was a hallucination. It was all a dream.

Steve: Anyway, I just wanted to point this out. It had never occurred to me, should've, that downloading, like using, like...

Leo: Oh, it's occurred to me. Only because how often do you do a Google search? You've got a doc, and you want to turn it into a PDF, or you've got, you know, a WordPerfect document.

Steve: Yup.

Leo: And how often does that happen?

Steve: It comes right up.

Leo: And in the old days I used to go out on the Internet and look for tools. Not anymore.

Steve: Yes. It comes right up in a search. How do I convert this? Oh, just click this link for a free document conversion. And you think, oh, good, I don't have to install another one of those stinky programs.

Leo: Exactly.

Steve: I just want to get it done because I only have this one thing to do.

Leo: What's interesting to me is that they still work. So it sounds like they're taking existing programs and modifying them.

Steve: Yeah.

Leo: They still do the job. So I guess that way you go, oh, good, I got the PDF. You don't think about it.

Steve: Yeah, when Boris asks to purchase your document conversation domain name...

Leo: For big bucks.

Steve: We've got some bitcoin here.

Leo: Just include your PHP code, please.

Steve: That's right.

Leo: Yes.

Steve: The top court in South Korea rejected Meta's final attempt to dismiss a $4.6 million fine. Five years ago, South Korea's privacy watchdog - we talked about this back then - fined Meta, this was back in 2020, for sharing the data of 3.3 million South Koreans with third parties without their permission or authorization.

Leo: Ugh. Ai ai ai.

Steve: Meta lost that battle. Then they appealed. They've now lost the appeal. The final highest court in South Korea said, no, we need some money. So they've got to pay.

Leo: Was it a breach? Or did they actually sell it?

Steve: It was actually sold. They were just saying, here's who were using us in South Korea.

Leo: See, this is - so for a long time I've said, oh, you don't have to worry because Meta's never going to sell your information. They sell - that's their secret sauce. They sell ads against that information. So they say, well, you want 35-year-old men in South Korea, we can deliver that. But to learn that they're actually selling that on...

Steve: Well, actually the article says "sharing." So...

Leo: Yeah, but...

Steve: Maybe not monetizing overtly. But, you know, like with their advertising partners; right? They want their advertisers to know as much about you as they can because we know that makes it a more valuable ad.

Leo: Yeah, but they don't - so for them to say, here's Steve Gibson's personal information is different than saying I will sell you an ad that will reach Steve Gibson and people like him. Because if you give Steve Gibson's personal information, well, who knows what Meta's up to.

Steve: Yeah. Anyway, what apparently the...

Leo: I'll have to adjust what I've been telling people is what I'm thinking.

Steve: The search into this said that Meta, without permission, five years ago...

Leo: Was selling them, not sharing them.

Steve: ...was sharing the data of 3.3 million South Koreans, enough so that they have just lost all of their appeals and are going to have to pay a $4.6 million fine. Which of course is a drop in the bucket for Meta.

Leo: Sure.

Steve: I mean, they're not - they have that in the petty cash drawer for the delivery guy when he comes up and...

Leo: But at least we now know they do that. That's the key to that.

Steve: Yeah, exactly.

Leo: Wow.

Steve: Okay. So Google has weighed in on their side of the Age Verification requirements. Google is, and speaking of Meta, Google is reported to be extremely upset over Meta's sponsorship, is the way Google phrased it, and their push for that Utah age-verification bill that we talked about last week, which moved through Utah's legislature. As we know, the Utah law transfers the responsibility of the task of checking, for example, a suspected child account from the application to the application provider, the store, essentially, offloading it, offloading the responsibility from individual apps, which is of course why Meta thinks that's a good idea.

Last week we looked at what Apple was doing, and last Wednesday Google posted their position about this under the title "Google's legislative proposal for keeping kids safe online." So they're calling it a "legislative proposal," meaning we're offering this to, you know, to the legislators as what we suggest people do. And in an indication of Google's annoyance with Meta, the tag line under that read: "Legislation pushed by Meta would share kids' information with millions of developers without parental consent or rules on how it's used; we have a better way."

So here's what Google said. They wrote: "Everyone wants to protect kids and teens online and make sure they engage with age-appropriate content, but how it's done matters. There are a variety of fast-moving legislative proposals being pushed by Meta and other companies in an effort to offload their own responsibilities to keep kids safe to app stores. These proposals introduce new risks to the privacy of minors, without actually addressing the harms that are inspiring lawmakers to act. Google is proposing a more comprehensive legislative framework that shares responsibility between app stores and developers, and protects children's privacy and the decision rights of parents.

"One example of concerning legislation is Utah's App Store Accountability Act. The bill requires app stores to share if a user is a kid or teenager with all app developers," they said, "(effectively millions of individual companies) without parental consent or rules on how the information is used. That raises real privacy and safety risks, like the potential for bad actors to sell the data or use it for other nefarious purposes. This level of data sharing is not necessary. A weather app doesn't need to know if a user is a kid." I'm still annoyed by the use of the term "kid," but okay.

"By contrast, a social media app does need to make significant decisions about age-appropriate content and features. As written, however, the bill helps social media companies avoid that responsibility despite the fact that apps are just one of many ways that kids can access these platforms. And by requiring app stores to obtain parental consent for every single app download, it dictates how parents supervise their kids and potentially cuts teens off from digital services like educational or navigation apps." Okay, I don't quite get that, but okay.

"By contrast, we are focused on solutions [we Google] that require appropriate user consent and minimize data exposure. Our legislative framework, which we'll share with lawmakers as we continue to engage on this issue, has app stores securely provide industry standard age assurances only to developers who actually need them and ensures that information is used responsibly. Here are more details." And we have a few bullet points.

"First, under privacy-preserving age signal shared only with consent," they write: "Some legislation, including the Utah bill, require app stores to send age information to all developers without permission from the user or their parents. In our proposal, only developers who create apps that may be risky for minors would request industry standard age signals from app stores, and the information is only then shared with permission from a user or their parent. By just sharing with developers who need the information to deliver age-appropriate experiences, and only sharing the minimum amount of data needed to provide an age signal, it reduces the risk of sensitive information being shared broadly." 100% agree.

"Appropriate safety measures within apps," they wrote: "Under our proposal, an age signal helps a developer understand whether a user is an adult or a minor. The developer is then responsible for applying the appropriate safety and privacy protections. For example, an app developer might filter out certain types of content, introduce 'take a break' reminders, or offer different privacy settings when they know a user might be a minor. Because developers know their apps best, they are best positioned to determine when and where an age-gate might be beneficial to their users, and that may evolve over time, which is another reason why a one-size-fits-all approach won't adequately protect kids."

Under "Responsible use of age signals," they wrote: "Some legislative proposals create new child safety risks because they establish no guardrails against developers misusing an age signal. Our proposal helps to ensure that age signals are used responsibly, with clear consequences for developers who violate users' trust. For example, it protects against a developer improperly accessing or sharing the age signal."

Under "No ads personalization to minors: Alongside any age assurance proposal, we support banning personalized advertisements targeting users under age 18 as an industry standard. At Google, this is a practice we've long disallowed. It's time for other companies to follow suit."

And finally, under "Centralized parental controls," they write: "Recognizing that parents sometimes feel overwhelmed by parental controls across different apps, our proposal would provide for a centralized dashboard for parents to manage their children's online activities across different apps in one place and for developers to easily integrate with." So they finish: "Google has demonstrated our commitment to doing our part to keep kids safe online. We're ready to build on this work and will continue engaging with lawmakers and developers on how to move this legislative framework for age assurance forward."

So, yes. If that sounds like a lot of what Apple was saying last week, it's a yes. I mean, when Apple and Google being the two gorillas in the market, they appear to be converging onto the same solution. Essentially, parents are able to group the phones of

their family members and indicate which phones belong to their minor children. Once this is done, children wishing to download applications with mature ratings will require parental consent. Developers of restricted apps have no need to know anything about those who are downloading and installing their apps. The fact that they're able to do so means that they have permission, either by using an adult's phone, or because a parent or guardian gave a child permission. So essentially, providing control only where it's necessary, which, you know, is very much like what Apple suggested and we talked about last week.

So it feels like that's where we're going. And it also feels like Google is rolling up their sleeves, calling this, you know, legislative proposal. So, you know, they're going to respond to legislation like what we just saw happening in Utah and say no, no, no, let's do it this way. This, you know, this is the way it should be. Unfortunately, our current administration seems upset with Google. I guess actually Biden's was, too. So...

Leo: Yeah, it was actually Biden's FTC that brought the...

Steve: That began the whole antitrust work.

Leo: Yeah, yeah.

Steve: Yeah. I got a kick out of this because I mentioned at the top of the show, Kazakhstan has a different approach. The Kazakhstan government has, get this, introduced SIM cards specifically designed for use of and by children. In Kazakhstan, all parents will be required to buy and deploy the new SIM cards for use in their children's devices. The cards come with built-in filters to restrict access to dangerous websites and social media. The cards also report a child's location to parents through a special app.

So, overall, it feels as though things are rapidly becoming a mess with random and uncoordinated legislation being created left and right. And frankly, I lay this at the feet of Apple and Google, who both resisted taking the action they could and should have taken on this many years ago. You know, they were like, no, no, no, no. We don't want any responsibility. We don't want any part of this; you know? And it's only when bad legislation and bad solutions are finally being created that now they're saying, oh, well, okay, yeah, what you're doing is wrong. Here's how we'll do it. So, you know, I guess, you know, better late than never.

Also, one last little bit. The Spanish government passed a bill last week to impose very stiff fines on companies that produce and dispense unlabeled AI-generated content. And when I say "stiff fines," we're talking up to 35 million euros...

Leo: Oh.

Steve: Yeah.

Leo: Wow.

Steve: That will get your attention, or 7% of a company's global annual revenue, whichever is greater.

Leo: Whoa.

Steve: The law intends to curb the spread of deepfakes and non-consensual adult content such as producing, you know, fake celebrity videos. Spain is the first country in the EU bloc to incorporate provisions from the EU AI Act into its national legislation. So they're saying we're going to fine you if you do not clearly label content as AI-generated.

Leo: I think that's reasonable. The fine's not, but I think...

Steve: We need it. We need it. But the fine...

Leo: It's a little outrageous.

Steve: The fine'll take your breath away.

Leo: Yeah, yeah.

Steve: Okay. We're going to talk about Google and the Canary after another break because we're now at an hour in, my friend.

Leo: The Google and the Canary. Wow.

Steve: Google and the Canary.

Leo: Sounds like a...

Steve: Maybe it's a reverse canary. I'm not sure if it's a reverse canary. We'll have to think about that.

Leo: Oh, that's a good point. No, it's - no, it's a canary.

Steve: It's a canary? Okay.

Leo: Well, we can talk about what the difference is.

Steve: Yeah.

Leo: Yeah. Well, we'll talk about it. Save - put a pin in it, as they say.

Steve: A canary is published. I'm thinking that a reverse canary...

Leo: Is the absence of something that is the canary; right?

Steve: Right.

Leo: So if you say in your legal disclaimers, "And we have never received a warrant from the United States government," and then it disappears, that's a reverse canary. Right, because without saying anything, you have said something. So Apple did a canary. I'm glad we get these things cleared up. You see, you don't just learn about security here. You learn about the use of the English language. But I do have...

Steve: This podcast is for the birds.

Leo: Literally. Okay, Steve. I want to hear about Google's Canary.

Steve: Okay. So last Friday, The Record ran a piece that caught my eye. In the wake of what has become an extremely public withdrawal of enabling Apple's strongest privacy guarantees for iCloud backup in the UK, many have wondered - including, it turns out, elected members of U.S. legislation - about Android and Google. What's their similar status relative to the United Kingdom, you know, their even larger Android ecosystem, which is designed and managed by Google?

Leo: I've wondered this, too. I figured if they went after Apple, I'm sure they must have gone after Microsoft and Google and everybody else; right?

Steve: Yes. The Record gave their coverage of this question the headline "Google refuses to deny it received encryption order from UK government," and apparently they've been asked directly and rather pointedly. The Record wrote: "Google has refused to deny receiving a secret legal order from the British government, according to a bipartisan group of members of Congress who are concerned Westminster may have demanded that several U.S. technology companies provide its security services with a mechanism to access encrypted messages. It follows the British government reportedly issuing such a secret legal demand, officially known as a Technical Capability Notice, to Apple. Apple is believed to be contesting the demand at a closed court hearing on Friday." And I assume they meant last Friday.

Leo: This most recent Friday the 13th, yeah, or the 14th, yeah.

Steve: "In a letter published Thursday [last Thursday], the members of Congress [U.S. Congress] complained about the secrecy of this court hearing, arguing 'it impedes Congress's power to conduct oversight, including by barring U.S. companies from disclosing foreign orders that threaten Americans' privacy and cybersecurity.' Despite widespread reporting of this TCN issued to Apple, the company [Apple] is prohibited from confirming whether it had received such an order under the UK's Investigatory Powers Act. In their letter, the members of Congress wrote that Apple had informed them 'that had it received a technical capabilities notice, it would be barred by UK law from telling

Congress whether or not it received such a notice.' Companies who have not received such a notice are obviously free to say so.

"The group wrote: 'Google also recently told Senator Ron Wyden's office that, if it had received a technical capabilities notice, it would be prohibited from disclosing that fact.' Experts, including from Britain's own intelligence community, have said that the government's attempts to access encrypted messaging platforms should be more transparent. Academics described the Home Office's ongoing refusal to either confirm or deny the legal demand as unsustainable and unjustifiable."

Okay. So what does this mean? I'm here to formally let everyone know who is listening to this podcast, know that I have not - I am not in receipt of any such or similar demand from the UK government. And Leo?

Leo: And I am not either, scout's honor.

Steve: Yes. I would imagine you are equally free, and now you have, you have said the same thing.

Leo: So that's conclusive.

Steve: Not that the UK government has any interest in either of us, or anything that we may have encrypted.

Leo: But we wouldn't be able to say anything had we received that. Including denying it, I presume.

Steve: Right. I could not, apparently, confirm or deny.

Leo: Actually, I bet you could deny it. But if you said I cannot confirm or deny, that's the reverse canary; isn't it. You could say...

Steve: Yes.

Leo: I mean, you could be lying.

Steve: So doesn't Google's refusal to simply say, as I just have, and as you just have, that they are not in receipt of an order which compels them to not disclose such an order, automatically mean that they are in receipt of a similar order from the UK?

Leo: I think that's a reasonable induction, I agree, yes.

Steve: And also wouldn't that also make sense? Wouldn't we expect Google to be just as much a subject of this as Apple?

Leo: Right.

Steve: And if Google were not, think about that. If the UK only required Apple to comply, wouldn't that constitute unfair meddling in the direct commercial interests of these two commercial platforms?

Leo: True, true.

Steve: Forcing Apple to be able to decrypt, to like publicly be able to decrypt the confidential and private information of their users, while not requiring exactly the same from others, would put Apple at a significant commercial disadvantage relative to its competitors. So that's not copacetic. It seems clear that whereas news of Apple's receipt of this leaked out, the same may have happened within Google, that is, the same receipt of this, but it hasn't leaked. You know, and of course some have suggested that Apple's leakage may have originated from within Apple itself as a means of opening this issue to the disinfecting light of day.

So, interesting. I think we have to assume that Google is also in receipt of this, and they're just, you know, they're not - they're like, you know, Sergeant Schultz. They don't know anything. They're not going to say anything. And I guess many of our listeners, our younger listeners, don't know what I'm talking about; but look up "Hogan's Heroes" and you'll find out.

And this brings us to another piece of related reporting from The Record, which they posted last Thursday, which was the day before this. Their headline was "Calls grow for UK to move secret Apple encryption court hearing to public session." The Record wrote: "Politicians and civil society groups in the United Kingdom are calling for a secret court hearing expected on Friday about the British government's encryption demands on Apple to be held in public. It follows warnings from experts, including from Britain's own intelligence community, that the government's attempts to access encrypted messaging platforms should be more transparent. Academics described the Home Office's ongoing refusal to either confirm or deny the legal demand as unsustainable and unjustifiable.

"The Schedule for the Investi" - why can I not say that? - "Investigatory Powers Tribunal, the only court in the country that can hear certain national security cases, includes a hearing set to take place behind closed doors on Friday [presumably last Friday] featuring the Tribunal's president, Lord Justice Singh, alongside the senior High Court judge Justice Johnson. It follows Apple disabling the option for its British users to protect their iCloud accounts with end-to-end encryption last month, in the wake of a reported legal order from the British government requiring Apple provide it with access to encrypted iCloud accounts. The hearing is purportedly the company's attempt to contest this order, although it is unknown on what legal grounds that attempt is being made."

So, like, you know, Britain has this law. They're saying, you know, commercial entities that we serve a secret order to must comply. So how does Apple say no? Maybe it's this competitive disadvantage thing I talked about. I don't know.

Anyway: "The British government continues to say it neither confirms nor denies the existence of such legal demands. Apple has not confirmed the reason the encryption feature was turned off, and would be prohibited from doing so."

Leo: Yeah, they can't say anything.

Steve: It's just nuts. This whole thing is nuts. This whole, you know, we're giving you secret orders that you can't ever talk about, but it's going to - but it requires that your behavior be modified.

Leo: But you remember, we've talked about it, we do the same thing. The Patriot Act sent - you can send - they send out National Security Letters, and you cannot say that we have received a National Security Letter and revealed all your information to the government. You can't tell anybody that.

Steve: I guess the issue here is that Apple cannot comply. And so if they're forced to comply, they're forced to change, to rollback their technology.

Leo: Right.

Steve: And so that's a big deal.

Leo: You know, and by the way, our own intelligence services, Tulsi Gabbard, the DNI, has said we have a treaty with England that says we won't spy on their people if they don't spy on our people. And this Investigatory Powers Act specifically said no encryption. We want to be able to read everything globally. Not just for UK citizens. We want to read Steve Gibson's stuff. And that's according to Tulsi Gabbard a violation of our own treaties with Great Britain.

Steve: Right.

Leo: So that may be where the argument goes in this - we'll never know because it's a secret court, as well.

Steve: "In a joint letter that was sent Thursday to the head of this, Lord Justice Singh, by a collection of British civil liberties groups, they asked him to use his discretion, because he had discretion, to open the hearing to the public, arguing that doing so would not prejudice national security. The campaigners for this issue said, they wrote: 'There is significant public interest in knowing when and on what basis the UK government believes that it can compel a private company to undermine the privacy and security of its customers.' They argued that there are 'no good reasons to keep this hearing entirely private' - it's probably embarrassment; right? - 'given that the existence of the secret legal order has been publicly reported and effectively confirmed by Apple's decision to remove its end-to-end encrypted service for British iCloud users.'

"Politicians from opposition parties, including the Conservative Party, Liberal Democrats, and Reform, have also called" - I mean, everybody wants more transparency from the Home Office. "David Davis, a Conservative Party politician who has long campaigned to limit state surveillance powers, told Sky News the government needed to explain its case to the public if it wants 'effectively unfettered access' to private data." So this is all good. This mess...

Leo: Secrecy is the authoritarian's friend. That's really [crosstalk], yeah.

Steve: Yes. And all of this mess, all this noise is what we need. You know, these decisions need to be made. And so I'm glad this is all, you know, coming to a head.

Leo: Yeah, me, too.

Steve: It's what we need to have happening because this all needs to be decided one way or the other. And, importantly, since the delivery of privacy and confidentiality is a commercial competitive attribute, whatever the rules finally turn out to be must be universally applicable to all parties equally and evenly. You know? And at this point, nothing about this process of secret UK government compulsion can become or remain the status quo. It has to change.

Leo: Mm-hmm. I agree.

Steve: Okay. So everybody with PHP-based servers listen up. Before we get to some feedback from our listeners, I want to make absolutely certain that anyone who's responsible for any PHP-based Windows web servers - so not those running Linux. This is not a Linux issue. I'm running Windows-based PHP servers at GRC, our web forums, our email system, the GRC.sc shortcut link redirector, all that's over on its own server because these are PHP based. That server is sequestered. It is on an isolated network that has no access to the rest of GRC because it's PHP, for exactly the reason I'm about to be telling everybody about. You know, I had the ability to do that, and since it wasn't code that I wrote, it's going to have its own little home where, you know, if it melts down, well, that'll be unfortunate, but it's got backups and rolling backups and everything. Still, I did not want it to be able to reach over into GRC.com and everything else that's there.

So the good news is that the several ways the PHP interpreter - and there it is, like, you know, interpreter, right, we know what a danger interpreters are. The several ways the PHP interpreter can be invoked, only the oldest original method of using the php-cgi.exe executable gateway, or frankly the php.exe itself, if it were to be placed in the php-cgi directory, is vulnerable.

Leo: Well, we've known this for years; right? I mean, this is not a revelation.

Steve: Well, CGI is not safe.

Leo: Right.

Steve: But the XAMPP system still uses it by default.

Leo: Oh. Oh.

Steve: That's what it's using. That's what it's using, you know, an oldie and goodie.

Leo: I remember putting an open file share on my server. This is many years ago. And what I didn't think - I thought people were going to upload files. Somebody did, and they uploaded a PHP file and executed it because I was running CGI, PHP-CGI, and any file in any folder could be executed if it's PHP.

Steve: Yup.

Leo: Big flaw. I learned a lesson then.

Steve: It's really bad.

Leo: Yeah.

Steve: So none of the newer approaches, including Mod-PHP, FastCGI - which is what I'm using - or PHP-FPM are vulnerable. However, as I said, on Windows the common use of the so-called XAMPP stack is vulnerable in its default configuration because it uses the php-cgi executable to invoke the PHP interpreter. You know, and XAMPP refers to the Apache web server, the MariaDB database, and both the PHP and Perl interpreters. So I breathed a personal sigh of relief at this, since all of GRC's many web servers have always been configured to use the FastCGI method of invoking PHP.

Before I talk about this further, the only solution is to move to the current release of a supported PHP, which means, if you're on the 8.1 track, the 8.1.29 or later; if you're using 8.2, be it 8.2.20 or later. I'm at 8.2.28 as of yesterday because of this news. I brought my servers up to speed because I was back on a vulnerable version. And it's, you know, it's easy to be. That was last summer. And the good news is I have FastCGI, so in this case I wasn't vulnerable.

But it's like, yikes. And if you're on PHP 8.3, be at 8.3.8 or later. And unfortunately, this still leaves a massive population of publicly exposed PHP servers vulnerable to complete system takeover. That is, I saw the command line. I'm not keeping it a secret, but it wasn't worth putting in the show notes, a command line that, when received by any of these vulnerable PHP systems, causes the system to reach out and download from an external server the content that they then want to execute on the vulnerable host. So, I mean, it is really bad. It's as bad as it could be.

Okay. So here's the backstory. The news that put me onto this was just published by The Record. They wrote: "Researchers said Friday" - and this is the point because this, as I said, this is about nine months old, but it's just ramping up. "Researchers said Friday that a vulnerability initially exploited mostly in cyberattacks against Japanese organizations is now a potential problem worldwide. The threat intelligence company GreyNoise said exploitation of the bug, tracked as CVE-2024-4577, 'extends far beyond initial reports,' referencing in particular a blog post published Thursday by Cisco Talos. The Talos team had said an unknown attacker was 'predominantly targeting organizations in Japan' in January through the vulnerability, which affects a setup called PHP-CGI that runs scripts on web servers. A patch was issued last summer.

"Cisco Talos said the attackers' apparent goal was to steal access credentials and potentially establish persistence in a system, 'indicating the likelihood of future attacks.' GreyNoise said it observed similar activity beyond Japan, revealing 'a far wider exploitation pattern demanding immediate action from defenders globally.'" That is, this

thing has just - it's recently exploded. Get this. "There are 79 known ways to exploit the vulnerability and remotely execute code on a compromised system."

Leo: I think we need Paul Simon for this, "79 ways to [indiscernible]."

Steve: That's right. And not only remotely execute code, but remotely execute code which you've induced your server to download for you.

Leo: Oh, wow.

Steve: I mean, it is really awful.

Leo: It's really bad.

Steve: "The PHP scripting language," they wrote, "is decades old and is widely used in web deployment. 'Attack attempts have been observed across multiple regions, with notable spikes in the U.S., Singapore, Japan, and other countries throughout January 2025.' Cisco Talos said Thursday that the attacker it studied used a command-and-control server that deploys a full suite of adversarial tools and frameworks." Why not download them all? I mean, this thing will let them download anything they want into a vulnerable server and then run them.

Leo: Oh, my god. Yeah, get them all.

Steve: It is just awful.

Leo: Put all 79 exploits on there.

Steve: That's right. "The researchers said they believed the attacker's motive was to move beyond just stealing credentials. Researchers at Symantec had reported exploitation of this CVE last August against a university in Taiwan, not long after the patch was issued."

The discovery of this is credited now, Leo, to an old friend of ours whom we have not heard much from recently, good old Orange Tsai.

Leo: Oh, yeah.

Steve: At DEVCORE.

Leo: Mr. Pwn2Own.

Steve: Uh-huh. In just the previous four years Orange Tsai has won, in 2021, 28th of Top 100 Microsoft Most Valuable Security Researchers award; in 2021, the Champion of Pwn2Own Vancouver. Also in that year a third of Top 10 Web Hacking Techniques for Exchange Server Remote Code Executions. He also won the Pwnie Award in 2021 for the "Best Server-Side Bug" for Exchange Server Remote Code Executions. The next year, in '22, he was the Champion of Pwn2Own Toronto. In 2024 last year, first of Top 10 Web Hacking Techniques for research of Confusion Attacks, and the fourth of Top 10 Web Hacking Techniques for research of WorstFit Attack. So we know the guy. I mean, this guy is a super hacker and a responsible researcher. Last June...

Leo: He probably makes a lot of money doing this, I imagine millions, yeah.

Steve: Yeah, yeah. Last June 6th, when DEVCORE published their Security Alert titled "CVE-2024-4577 - PHP CGI Argument Injection Vulnerability," it drew the security industry's attention. They opened with: "During DEVCORE's continuous offensive research, our team discovered a remote code execution vulnerability in PHP. Due to the widespread use of the programming language in the web ecosystem and the ease of exploitability" - I mean, this thing is drop-dead simple to exploit, and that's one of the big concerns. This is script-kiddie heaven. "DEVCORE," they wrote, "classified its severity as critical, and promptly reported it to the PHP official team. The official team released a patch on 6/6. Please refer to the timeline for disclosure details."

And I'll interrupt here just to say in their published timeline we see the way this is all supposed to go. For one thing, the PHP developers well understood the nature of critical bugs. You know, can you say "interpreter"? I mean, they've had their hands full for decades dealing with PHP interpretation bugs. And secondly, they all know Orange Tsai and DEVCORE. So when you get a universal scope bug report marked "CRITICAL" from these guys, your plans for the next several days, if not weeks, just changed.

So the timeline says on May 7th, DEVCORE reported the issue through the official PHP vulnerability disclosure page. That same day, PHP developers confirmed the vulnerability and emphasized the need for a prompt fix. Nine days later, on May 15th, PHP developers released the first version of the fix and asked for their feedback. Two days later, on the 18th, the developers released the second version of the fix and asked for additional feedback. Another two days later, PHP entered the preparation phase for the new release version. That was May 20th. And then on the 6th of June, the next month, PHP released new versions 8.3.8, 8.2.20, and 8.1.29.

Under "Description," the DEVCORE people - so we're back to the DEVCORE disclosure now. Under their description they explained: "While implementing PHP, the team" - meaning the PHP team - "did not notice the Best-Fit feature" - get this, Leo, you're going to love this bug, oh, my god - "the Best-Fit feature of encoding conversion within the Windows operating system. This oversight allows unauthenticated attackers to bypass the previous protection of CVE-2012-1823 by specific character sequences. Arbitrary code can be executed on remote PHP servers through the argument injection attack." In other words, this PHP bug was originally found and fixed 13 years ago.

Leo: Wow.

Steve: Back in 2012. But Windows employs its own "best-fit" UNICODE character conversion feature, and Orange Tsai discovered that many, apparently 79, other deliberately crafted UNICODE character sequences would be transliterated by Windows on the fly and used to bypass the fix from 2012. So this vulnerability had been there

since 2012, never repaired, as it was believed to have been and was under Linux, because Windows just changes characters as it wants to.

Leo: To whatever the best fit would be.

Steve: That's right. You didn't really mean that. You meant this.

Leo: You wanted the better fit.

Steve: It's a better fit, yes. And, oh, whoops, it bypassed a fix that we put in to prevent that from happening.

Leo: Twelve years ago. Wow.

Steve: Yeah.

Leo: Wow.

Steve: This thing is so bad, for example, that a single query issued to any vulnerable Windows web server can cause, as I mentioned, to fetch any remote file named in the query and then execute that file, no matter what it might be, on the vulnerable machine. That's not anything that anybody wants to have happen on their server. Under the "Impact" section of their disclosure, they were very clear. They wrote: "This vulnerability affects all versions of PHP installed on the Windows operating system." Period. All of them.

They also noted: "Since the branch of PHP 8.0, PHP 7, and PHP 5 are End-of-Life and are no longer maintained anymore, server admins can refer to the 'Am I Vulnerable?' section" - and the answer just is yes - "to find temporary patch recommendations in the Mitigation Measure section."

And in that "Am I Vulnerable?" section they wrote: "For the usual case of combinations like Apache HTTP Server and PHP, server administrators can use the two methods listed in this article to determine whether their servers are vulnerable or not. It's notable to address that Scenario-2 is also the default configuration for XAMPP for Windows, so all versions of XAMPP installations on Windows are vulnerable by default.

"As of this writing, it has been verified that when Windows is running in the following locales, an unauthorized attacker can directly execute arbitrary code on the remote server." And so they show Traditional Chinese using Code Page 950, Simplified Chinese using Code Page 936, and Japanese using Code Page 932.

"For Windows running in other locales such as English, Korean, and Western European, due to the wide range of PHP usage scenarios" - in other words, it was just too much for them to check, you know - "it's currently not possible to completely enumerate and eliminate all potential exploitation scenarios. There are just too many to fix. Therefore, it is recommended that users conduct a comprehensive assessment, verify their usage scenarios, and update PHP to the latest version to ensure security." And, you know, even

though I was using a non-vulnerable fast-CGI implementation, I'm not taking any chances. So I did move to the latest version yesterday.

That was written last June. Since then it's been widely confirmed that this vulnerability can be exploited anywhere and on any vulnerable server regardless of local language configuration. Therefore, by far the safest and most recommended mitigation is to update to a version of PHP that once again fixes this problem. You know, assuming that you have 8.1.2 or .3, it's just a sub-version update. So it should be as simple as just dropping new binaries into the existing PHP directory. And then you're good to go.

So, you know, it should be a simple fix. But I just, I wanted to absolutely be sure because this thing is so bad, and it is so likely that many default configurations will be vulnerable, and the exploitation of this is ramping up, you know, very, very quickly. So I want to make sure all of our listeners know, and anybody that they know that may be running PHP on Windows servers, only Windows that is the problem because of Windows Unicode. It's Unicode that is doing this...

Leo: Oh, it's that best-fit thing.

Steve: ...best-fit character translation nonsense.

Leo: Yeah.

Steve: Which essentially created a workaround on behalf of the attackers for the fix that had been implemented back in 2012 when this was first found.

Leo: Amazing.

Steve: Okay. I need to take a...

Leo: Please.

Steve: Catch my breath and sip some coffee, and we're going to talk - we're going to look at listener feedback next.

Leo: I think we all need to catch our breath after that, actually. Geez. I'll never forget that. It must have been the very early days of the show. I think I was giving people a place they could upload something to the server. So I had an open file share. What I didn't understand was that somebody could upload plaintext.php file that could then execute.

Steve: Yup.

Leo: Fortunately I think we caught it before it got...

Steve: I think I remember you talking about it on this show, too.

Leo: Yeah. It was like, that was quite an eye-opener, I guess, you know, PHP can be executed from, if you're using the CGI, from any folder anywhere unless you specifically lock it down. We learn; right? That's the whole point of the show. That's the whole point of being human. We make mistakes, and we learn. I hope we've learned now that we'd better back up our data and make sure we have copies of it. All right, Steve. I hope you are thoroughly refreshed, that you're now in the...

Steve: Ready to go. Next phase.

Leo: What do they call it, the back quarter of the show, or I don't know what they call it.

Steve: So our listener Sam Miorelli wrote: "Hey, Steve. On the applications thing" - meaning employees from North Korea - he said: "I run an industrial cybersecurity business. Last year, before we all knew about these things, we got an applicant (who we hired to work in person), who was incredible on the CV (lots of certs, including for FortiGate) and video interview. We foolishly ignored warning signs when the in-person manager first met him post-offer and pre-start, and things seemed a bit off. After he started, it was immediately clear the CV didn't reflect his actual skills. You know, he was googling how to apply firewall rules on modern GUI firewall admin interfaces.

"When I endorsed him, I chalked up his strange conversation style during the video interview to be from his accent, you know, and cultural as he's from India. And he had all the right answers. And, wow, again, what a great CV. In hindsight, I'm convinced he was using an AI interview helper tool like @finalround_ai." Which I hadn't heard of before, @finalround_ai. Sam said: "Of course, it's impossible to prove these things, so we're having to think harder about how we screen applicants in the future. Lots of phonies out there, not just the North Koreans."

Leo: Wow.

Steve: So a little bit of feedback from one of our listeners.

Leo: And a tip on the AI you might want to use for your next job.

Steve: That's right. If you happen to be interviewing, want to sound a little more polished.

Leo: Yes.

Steve: Ian Beckett said - actually these were a couple of tweets: "@SGgrc re: SN-1012," our episode, he said, "and Microsoft's Sysinternals tools." He said: "These tools are so popular, it's astonishing Microsoft's engineers don't securely recode these tools. The little Sync Toy tool," he said, "(download now removed from Microsoft's Sysinternals site), still provides just about the only way to simply do a regular Windows sync backup to external

drives using a TRUSTED tool. The pitiful inbuilt Windows 11 backup tool's only purpose is seemingly to drive revenue to OneDrive subscriptions." He said: "I really despair of Microsoft nowadays. Unless it generates online services revenue, they have little interest in user experience."

And Ian is, of course, referring to the DLL injection vulnerabilities that were recently discovered to adversely impact the security of the use of Sysinternals tools. Rather than loading the standard system DLLs from the system's well-known directories, the tools have retained Windows' once deliberate, though extremely insecure design of first looking in the executable's own execution directory before looking elsewhere. This allows bad guys to drop their own malicious versions of these DLLs, perhaps even older versions of Microsoft's own signed Windows DLLs that contain long since patched vulnerabilities, allowing them to effectively turn back the clock to be exploited again.

Microsoft reportedly said "tough beans." We're not planning to fix them, they said, which seems irresponsible. And as we noted at the time, frankly, even if they were fixed, there's still a massive inventory of them already deployed out in the world. And they never receive updates of any kind. So it's a mess.

TycoonTom tweeted: "@SGgrc: Hi, Steve. What's that networking app that shows you net traffic? The company was from Australia?" He was just, you know, he heard me referring to it.

Leo: I've got it running on my Mac right now.

Steve: It's a win, isn't it, Leo.

Leo: Yeah, I love it, yeah.

Steve: Yup. It's NetWorx, N-E-T-W-O-R-X, from a company called SoftPerfect. I've got a link in the show notes for anyone. It's free for 30 days, after which I would be surprised if you don't want it forever for 15 bucks. As I've noted, it will easily monitor the local machine. But my favorite feature is that from a local machine it's also able to monitor the real-time usage of the entire network by watching the router's SNMP interface byte counters.

Leo: Oh, nice. Oh, I forgot about that.

Steve: Yeah, yeah. You're able to set...

Leo: Oh, I've got to do that. That's great.

Steve: Yes. You set it up, you're able to set it up to monitor your entire, like, family or local network LAN use.

Leo: I need to do that, yeah, very nice.

Steve: Very cool.

Leo: Very nice. Good recommendation, thank you, Steve.

Steve: John David Hickin wrote: "I'm not even sure it deserves a CVE." Oh, he's talking about the backdoor, the so-called backdoor from last week. "I'm not even sure it deserves a CVE. This may well be similar to the case of the WIN32 API (and it's a DLL) versus the at least at one time undocumented API of NT.dll." He said: "These ESP32 undocumented commands may not be guaranteed to survive the next chip redesign. Device driver writers beware. Cheers, John."

Now, John's of course talking about last week's "Backdoor" that wasn't a backdoor. As we said, they were some undocumented functions in the SOC, the S-O-C, the System On a Chip hardware. And he's 100% correct that no one should be relying upon them for their own code since, being unofficial and undocumented, the Chinese chip maker Espressif should feel free to change their function or remove them entirely at any time. And I also agree that even assigning a CVE in retrospect was ridiculous, though I understand the discoverer's motivation behind doing so. You know, they were advertising this as a big bad backdoor, which was the narrative that most of the tech press picked up on this. So, yeah, you've got to have a CVE to make it sound more real and scary.

Mark Goldstein wrote: "Thanks for sharing Roger Grimes' story on the North Korean hackers. You did an important public service. The recitation of the story was funny and compelling podcasting." He says: "I told Roger of your recitation."

Leo: Oh, good. Nice.

Steve: Yup. And Mark said: "In 2009, I wrote a business plan for my company, America Online, to acquire LastPass."

Leo: Oh. Wow.

Steve: He said: "The CEO said we were not in the security business," meaning, you know, AOL was not. "So my proposal was shut down, although one day I visited Joe and his team with dozens of ice cream sandwiches on a hot Washington, D.C. day." Mark wrote: "After the first breach at LastPass I searched for a new password manager. I read what cryptologists said. I read FAQs and everything on various password manager websites. Finally, I found that 1Password had written some technical papers including their security model. It explained their various security choices. I could not evaluate all the crypto, but I understood their perspective of the vulnerabilities of password managers.

"I discovered that they knew users of 1Password could create easy-to-crack master passwords, so they used the master password along with a strong certificate to create the security for each instance of the password manager on a PC, Mac, iPhone, et cetera. When I create a new instance of 1Password, it copies the strong certificate to the new device. If someone cracks my 16-character password, they still must crack the 64-bit certificate. Good luck." And he finished, writing: "This is why I chose 1Password. Subsequently I use 1Password on my iPhone and Windows PC. Their cross-platform implementation of passkeys works great for me. Passkeys on 1Password is my security solution. Regards, Mark."

And I should mention that 1Password is also a sponsor of the TWiT network. And I wanted to thank him for sharing his note and experiences. And many of us agree that that's a great, you know, that 1Password is doing a terrific job. I should note that I've always also been a fan of 1Password's additional user-account entropy which they introduce using a client-side blob. While it means that it must be duplicated and replicated across all of a user's devices, you know, that's a one-time requirement that then creates and provides very strong additional enduring security forever, which makes sense to me.

Leo: Yeah, we've talked about this before. And I remember I asked you is it more secure, and you said, well, if you use a good password, it's not. But just as Mark says, it's for people who use monkey123. But then it makes me wonder, well, what do you need the password for? You've got the certificate.

Steve: You've got - yes, right.

Leo: You know.

Steve: It's very much the way you and I also use...

Leo: Belt and suspenders.

Steve: You and I use certificates for SSH login.

Leo: That's right. That's right.

Steve: Because so it's both a password to say this is who we are, and a certificate so that, if somebody else tries to spoof who we are, you know, they can't get in.

Leo: I don't actually, once I have the certificate set up, use the password anymore. I just automatically log in.

Steve: Yes, it's super, super strong.

Leo: Because the key exchange, yeah.

Steve: Yup.

Leo: Yeah.

Steve: An anonymous listener said: "Steve, please keep my name confidential." He said: "I would like to explain to you what happened to LastPass a few years ago. I work for a major cloud distributor, and this occurred during a meeting with their CTO at the time,

since LastPass was one of our vendors. I asked what happened, and the CTO explained that the Dev at home was using Plesk on his personal Mac which was hacked due to a Plesk media server that had not been updated. That much we know."

He said: "But the primary issue was that he was logged into the LastPass network from his personal machine. I asked the CTO why he was able to log into LastPass's network from a personal machine since they had policies in place to prevent that. The CTO confirmed that they did not enforce their own policies. Also, the secret AWS keys where they stored their customer vaults was kept in LastPass Corporate Secure Notes, so was readily accessible to anyone." Wow. Even those who didn't need access to them.

> **Leo:** And of course as everyone knows Plesk is written in PHP, so it's doubly insecure.

Steve: Oh. "So your evaluation," he said, "of the product wasn't wrong. It's a good password manager. But the company itself was not well managed. Regards."

So there's a little bit of additional insight that we haven't had previously. Since we cannot know how and where crucial decisions were being made, there's really no way to assign specific blame. But one thing we do know is that LastPass really dropped the ball on the PBKDF iterations issue. And there's really no excuse for that. They just didn't care. We know that because once this was brought to the glaring attention of the industry, then they went to the trouble of autonomously updating everyone's iteration counts later, you know, retroactively. This proves that they could have done so at any time, but never had bothered to before.

As we know, I always draw a sharp distinction between policy decisions and mistakes. The LastPass developer whose machine was doubtless targeted and compromised was not practicing good security hygiene. And LastPass was not managing the connections to their corporate network securely. So the developer made a bad mistake. But not bothering to ever retroactively update original or older PBKDF iteration counts as a policy mistake. It wasn't a priority decision, like, to fix that, as it should have been. And that's unforgivable. That they need to be held accountable for. And it's only those people whose LastPass vaults are being cracked retrospectively, retroactively, essentially, because they had zero iterations or some, you know, 500, you know, low early iteration count. And that is all on LastPass.

Jeff wrote to us: "Steve, Mandiant is reporting on an espionage campaign by China, exploiting Juniper big-iron routers." And he provided a link to that from Mandiant which, you know, is the Google-owned security firm. And he cites it, saying: "End of life hardware and software. Yeah, that's a thing I see all the time." He said: "You don't want to know what I found on the network of my Fortune 500 defense employer last week. It's a bit of a dog-bites-man story, but it's part of a pattern by China to infiltrate critical infrastructure and hold it at risk as part of their national strategy. Signed, Jeff." He says: "P.S. Ha! I forgot to use my GRC-registered email. I appreciate the instant bounce, since I could fix that and resend in less than two minutes."

Okay. So since Jeff referred to his Fortune 500 defense contractor employer, I left off his last name, though it's familiar to me since he's been an avid provider of feedback through the years. I was familiar with the news that he linked to. Older Juniper routers have problems that have been resolved in later devices. And those older routers are no longer receiving updates. So they're stuck running older firmware that will never be repaired. Still, those routers are well built and running, so it's difficult for any CIO to tell his CFO that, you know, we need some money, and a bunch of money, to replace some aging network infrastructure equipment. You know, the CFO replies, "Okay. What's wrong

with it? Isn't it still working?" And our responsible CIO says, "Well, yeah, but it's old, and it's no longer being maintained by its manufacturer. So it could have some security weaknesses that could possibly be remotely exploited by foreign hostiles."

And the CFO says, "So you're saying that as far as you know there's nothing wrong with it, and it's still working just fine. But there might or might not be something wrong with it, and we wouldn't know?" And our CIO, feeling that he's losing this one, says: "Yes, that's exactly right. We could be in danger." And the CFO ends the discussion, saying: "Okay, I get what you're saying here. I really do. But, you know, we have some very, very pressing needs, and they're not what-ifs, they're real. It only makes sense for those to take priority."

So I don't know how this changes over time. Certainly every one of the C-suite executives appreciates the need for proactive security. That CFO would not blink at the need for an industrial-strength firewall appliance to keep the bad guys out if they didn't have one, and I'm sure there was one from the get-go. And I'm sure that intellectually everyone also appreciates the need for security updates and patches. Everything around them is constantly being updated and patched and fixed, their phones and their PC and now even probably the cars they drive. And we're all being told that these measures keep problems from ever occurring. But we never actually see any of these supposed problems; right? So they remain intangible, and it makes it a little difficult to sell.

It feels like this is going to require a cultural change, and that's just going to take time. And while I intensely dislike the "rental model," as we know, you know, that the world is moving toward, in the case of keeping older gear secure, there's real value being offered. Where I believe that, for example, Juniper has missed a trick is in choosing to allow their appliance, their older appliances to fall out of maintenance and to not tie its continued operation into an annual paid maintenance agreement. They're leaving money on the table by not keeping their older - by not offering to keep their older devices alive and maintained in return for some cash.

The very many companies with older and still working Juniper gear, they're not upgrading to newer devices because the older devices their customers already have are still working. But those customers do truly need security maintenance for those devices going forward, and they would probably pay for it if they were allowed to, but they're not. They're being told, oh, you've got to, you know, it's obsolete. It's old. It's no longer being maintained. You've got to buy new stuff. And it's not cheap. It's a lot more expensive than it was when they bought the first stuff. So why abandon a customer and their ongoing need for security? To me it makes no sense. But that's the way the business is happening.

Bruce Olson said: "I wanted to make sure you knew about this claim being made by users on Reddit. It seems that the organization behind ZimaBoards" - and that company is called IceWhale - "may be selling user information as some folks have started receiving marketing targeted at email accounts given to ZimaBoard." And he finished: "That's all I had to say. Thanks for all the great work, and always looking forward to the next episode. Bruce from Michigan."

So that's disappointing; right? It's certainly a reason for using an email aliasing service so that this abuse can be controlled by the email's recipient. And in the case of IceWhale, the ZimaBoard creators, I guess I can't say that I'm surprised. I receive a great deal of promotional email with all manner of special offers and come-ons from them, like directly from them. And I just went over to their site, and the top of the page has a bright orange scrolling banner saying, "Sign up now and unlock up to $50 for new members."

You know, I mean, so they're very promo happy over there at IceWhale. And if this concerns you, this argues for purchasing their boards through Amazon, which you can

do. But I suppose I would just chalk it up to the cost associated with obtaining a perfect little single board PC having two network interfaces, two SATA ports, a PCIe expansion slot, and Linux preloaded, all for 90 bucks. Ninety USD, and you've got this perfect little machine. It's still the best deal around, even if one does need to give them a temporary throwaway email address.

And what was freaky is that I did not plan this. As I was moving through my email feedback, the next note that popped up after Bruce's note about IceWhale selling our contact data was this note from Bill Allen with the subject "Loving my ZimaBoard!" And I've got two pictures that Bill included with his email in the show notes. He wrote: "Steve, I got started with a ZimaBoard specifically to run SpinRite more easily on hard drives in my office, which it does very, very well." Of course it would because it's what I used to develop SpinRite 6.1.

Leo: And it's got a SATA port, so you just connect it right - and it can run FreeDOS?

Steve: It's got a pair, a pair of SATA ports, yeah.

Leo: Yeah, yeah. I was going to say, I mean, I'm not sure it's better than the Raspberry Pi, which is 35 bucks. But that is how it's better. It's got a SATA port, yeah.

Steve: Well, and it'll run SpinRite, and a Raspberry PI won't.

Leo: Won't, right, exactly, yeah.

Steve: Right.

Leo: Is it an x86 architecture? It must be.

Steve: The ZimaBoard is, yes. It is Intel-based, yeah.

Leo: Interesting.

Steve: Anyway, so he said: "But the ZimaBoard has turned into a bit of an obsession, and a really fun project platform." He said: "Here is my ZimaBoard system." And he showed us a picture of it all wired up, and another picture of a screen. He says: "To its right is an outboard PCIe card carrier for the NVMe M.2 drive it's booting from." And he said: "Upper left is a mini travel wireless router in client mode." He said: "Down and to the left is an AdderLink IP KVM which is giving me keyboard, mouse, and video access to it across my local network via its internal VNC server. Currently running FreeDOS, as shown in the other photo. That FreeDOS install also has SpinRite 6.1 on it, of course." He says: "Thanks for pointing us to the ZimaBoard. Best Regards, Bill in Crowley, Texas."

So anyway, I've received many similar reports through the years since my discovery of this lovely little device. It's not super powerful. I always purchased the smallest of the three available models since it was just going to be running FreeDOS which, you know,

can be powered basically by a squirrel cage. But these little boards are the machines that built and tested SpinRite. So anyway, I just thought I would share that fun bit of feedback. It is a great little solution.

Mark Jones wrote a note that has some detailed lead-in, but I loved his story, which is a bit of a head shaker. So the subject of his email feedback was "AI and Microsoft Defender." Get this. Mark wrote: "Dear Steve. Love the show, loyal listener since Episode 1, Club TWIT member. I really appreciate you and Leo.

"I encountered something new that illuminates some of the comments you've made recently about AI. I volunteer with an organization that has websites and a newsletter. About half our membership is employed by one of two big multinationals. Both are Microsoft shops. Both have lots of barbed wire wrapping their IT infrastructure. Microsoft Defender blocks questionable sites. The sieve is set pretty tight. At one point when I was still working there, GRC.com got blocked."

And I'll just insert a little note here: For many years I was hosting known viral code for research purposes at GRC.com. The page contained, you know, the various archives and was very clearly marked as, you know, download at your own risk. Everything was red and flashing. And, you know, it was very clear that this was, you know, old viruses that people might want to play with. But any search engine or trawling bot sees ZIP archives containing known dangerous viruses and freaks out. So since there is no interest in that really anymore, that's long since removed, and some of those false positives that others were also reporting have ceased.

Anyway, Mark's note continues: "I moved 25 years' worth of our organization's newsletters to its own site three years ago. The site is only three PHP files, some XML for SEO, and a bunch of PDFs. I made the move after consultation with IT folks at the company I used to work for prior to retiring. They indicated that simpler was better at keeping out of the crosshairs of security sites. Sites that allow visitors to upload files are particularly troublesome to the corporate IT folks; and our main site, over my protests, has WordPress plugins that accept uploads.

"Just recently the site" - and he's talking about his site, MidlandChemist.org - "started being blocked by the corporate Microsoft protection." Meaning of the company he used to work for, which is using Windows Defender. He said: "I went to an IT friend and asked how I could fix it. After three years of being okay, the site was suddenly being blocked. He was kind and connected me with someone responsible for the blocking. Here is where AI comes in. Get a load of this! The filters" - meaning Microsoft Defender filters - "are now AI-based, not rules based. He could not tell me why the site was being blocked because there was no rule being tripped. There are no rules anymore. Something about the site triggered the AI algorithms. No reason could be given. It was just AI.

"Just as you described, AI makes connections that may elude human interpretation. The good news is there is a way to whitelist sites, provided I can find an employee willing to take responsibility. Regards, Mark."

Wow. You've got to love that one. We turned all site blocking over to AI, so it just does whatever it does. We no longer know what or how. Welcome to the future; you know?

Leo: So this is the Defender that everybody has on their Windows machine; right?

Steve: Yup.

Leo: Wow.

Steve: Yup.

Leo: Interesting.

Steve: A listener who just uses his initials, PV, said: "Steve, I was recently casting a line out into the sea of Kindle Unlimited suggestions. Unfortunately, I also ran into the 'Artifact' book you talked about before. But I also found a winner. The series is called 'Dumb Luck and Dead Heroes' by Skyler Ramirez. It starts out a bit rough in the first book. Both main characters are at a very low point in their lives, and there's a lot of wallowing in that. But it picks up really fast, and there's a lot of crazy fun space adventure and just the right amount of humor." And I thought of this because I know that our listeners enjoy books that incorporate some humor.

And he said: "Besides the main books, he has a lot of little side stories that are the strange-but-true details behind one of Brad's stories. And there's also three books about his 'best friend who's also a king's cross assassin,' which are a bit of a different tone, but fun, as well. I generally am not a fan of side stories, but I enjoyed all of these. To 1100 and beyond. Signed, PV."

So anyway, I appreciate, and I am forwarding, PV's recommendation without any of my own review. So I can't vouch for, and I'm not vouching for, "The Dumb Luck and Dead Heroes" book or series by Skyler Ramirez. But it's got some humor in it, and I just wanted to let our listeners know, if they're looking for another one of our listeners' recommendations.

While we're on the topic of sci-fi reading, for my part I am remaining ever-more-deeply hooked on Neal Asher's novels. I'm now into the third of the first five-novel "Agent Cormac" series. And toward the end of the second one I realized that I was really having a good time. As I've mentioned, I am super-finicky about the quality of writing, and these are fully satisfying for me in that regard. And he's building up some really interesting characters. You know, it's still pulp. I'm not meaning to suggest otherwise. And it's not free. Unlike PV's discovery of those "Dumb Luck and Dead Heroes" novels which he found through Amazon's Kindle Unlimited, these Neal Asher novels are $7 each.

But as we've said, with a five-shot Starbucks Latte now at $9.50, I am easily obtaining more than $7 worth of entertainment from each of these. And given how much Asher has written, and the comments online that they only get better and better with time, and I'm going back to the beginning and starting from there, I know I'm going to be stuck reading everything that he's written for quite a while.

And lastly, before we get to today's main question of just how susceptible any of the PC-compatible machines you may have may be to Rowhammer attacks, and while I'm reviewing sci-fi stuff, there's something Lorrie and I watched and immensely enjoyed last Friday evening. If someone who knew I had a subscription to Apple TV and that I enjoyed science fiction themes, if some such person were to recommend "The Gorge" to me, having just watched it Friday night, I would have been appreciative of their recommendation. So having seen and enjoyed the movie immensely, I am hereby making that recommendation to our listeners.

As the movie unfolded, it had all the promise of being what I call "a perfect movie." And there aren't many of them. They're rare. And this is not one, as it turned out.

Leo: Oh. You got my hopes up.

Steve: Well, about a third of the way through I said to my wife, "So far, this is a perfect movie." And by that I mean, you know, it's not going to win any awards. But as the plot unfolded, the movie was perfectly paced. It was in no hurry to get where it was going. You had no idea, you could not guess what it was about, even. I mean, it was a mystery for the viewer. It unfolded gradually. Only necessary facts were revealed. Also, it happened to star that actress who played the chess prodigy in "The Queen's Gambit."

Leo: Yeah, Anya Taylor-Joy.

Steve: Really like her.

Leo: Yeah, she's very...

Steve: Big eyes, very easy on the eyes. She was one of the two protagonists. Okay. So I have to say that it got a bit ridiculous, like maybe they were trying to create a videogame tie-in in the latter part of the movie. But having said that, I could easily watch the entire first portion of the movie again. I mean, it was so satisfying. And I imagine that a lot of our listeners may be a little less finicky about, you know, people who never die, despite how many shots are fired at them, that kind of thing. But okay. Still, I'm no longer 14, and I'm not a fan of implausibly ridiculous over-the-top violence. But it's there on Apple TV. If you're a subscriber, you already have it waiting for you. And I do recommend it. It was, you know, it's not, as I said, it's not an award-winner. But it was really enjoyable. And the first half was - it was perfect.

Leo: Yeah.

Steve: It was really good.

Leo: Good. I'll have to check it out. Now, back to Steverino because I'm dying to find out what's going on here.

Steve: It's rare that we're able to invite the listeners of this podcast to actively participate themselves in cutting-edge security research. But this week a research team that has been looking into and questioning the actual dangers presented by Rowhammer attacks is asking for as much breadth and depth of real-world participation from the field as they can get. This amounts to downloading an ISO file, writing it to a thumb drive, then booting and running the Arch Linux OS and Rowhammer data-gathering tests that it contains. I immediately downloaded the 1GB ISO, used the latest, for me, RUFUS v4.6 for Windows to transfer that ISO onto a 32GB thumb drive, booted it on my ZimaBoard, and let it run in the background while I worked on the podcast. Okay, but let's back up a bit.

We've been talking about the many various aspects and versions of the original discovery known as "Rowhammer" since its first description back in 2014. It was 11 years ago that this was first found. The essence of the problem is that in the inevitable quest to increase the density of main system dynamic RAM, you know, the RAM that's typically measured

in tens of gigabytes, engineers squeezed every last bit of noise margin out of their designs. The RAM still worked. Systems booted and for the most part ran reliably. But then some clever researchers came along and asked a question no one else had before. They asked: "What if we were to hammer over and over and over on one row of RAM or on the RAM on either side of one row? Might that confuse the nearby bits?"

And we know the answer to that question. It turned out that, yes, indeed, not only can neighboring bits be affected, but those effects can be powerfully weaponized to completely collapse and bypass the security boundaries and guarantees upon which all modern computing relies for its operational security.

During the decade that followed since 2014, these surprisingly prevalent and successful attacks have been elaborated upon and expanded by many groups of researchers across the globe. The attacks have been strengthened. As Bruce Schneier reminds us, attacks never get worse, they only ever get stronger. They've been optimized. They've been sped up. Researchers have even demonstrated web-based exploitation via JavaScript code and even using network packets, the receipt of network packets to induce Rowhammer vulnerabilities. And after the industry reacted to the initial news of these exploitable weaknesses with improved designs, you know, like DDR3 was where we were then. DDR4 was supposed to fix it, but didn't. DDR5 was supposed to fix it, but still hasn't. The industry reacted, trying to fix this. New designs, faster refresh, detection of Rowhammer attacks on the fly.

Anyway, nearly four years ago, in May of 2021, Google's security blog posted "Introducing Half-Double: New hammering technique for DRAM Rowhammering bug." Google's summary of their discovery is worth a quick review since it nicely lays out today's situation. They wrote, and so this was six years downstream from the original revelation of Rowhammer. They said: "Today we're sharing details around our discovery of Half-Double, a new Rowhammer technique that capitalizes on the worsening physics of some of the newer DRAM chips to alter the contents of memory.

"Rowhammer is a DRAM vulnerability whereby repeated accesses to one address can tamper with the data stored at other addresses. Much like speculative execution vulnerabilities in CPUs, Rowhammer is a breach of the security guarantees made by the underlying hardware. As an electrical coupling phenomenon within the silicon itself, Rowhammer allows the potential bypass of hardware and software memory protection policies. This can allow untrusted code to break out of its sandbox and take full control of the system.

"Rowhammer was first discussed in a paper in 2014 for what was then the mainstream generation of DRAM: DDR3. The following year, Google's Project Zero released a working privilege-escalation exploit. In response, DRAM manufacturers implemented proprietary logic inside their chips that attempted to track frequently accessed addresses and reactively mitigate when necessary. As DDR4 became widely adopted, it appeared as though Rowhammer had faded away, thanks in part to these built-in defense mechanisms. However, in 2020, the TRRespass paper showed how to reverse-engineer and neutralize the defense by distributing accesses, demonstrating that Rowhammer techniques are still viable." And we did a podcast on TRRespass. "Earlier this year, the SMASH research went one step further and demonstrated exploitation from JavaScript, without invoking cache-management primitives or system calls.

"Traditionally, Rowhammer was understood to operate at a distance of one row. When a DRAM row is accessed repeatedly, the 'aggressor' bit flips were found only in the two adjacent rows, the 'victims' on either side. However, with Half-Double, we've observed Rowhammer effects propagating to rows beyond adjacent neighbors, albeit at a reduced strength. Given three consecutive rows A, B, and C, we were able to attack C by directing a very large number of accesses to A, along with just a handful, dozens of flips, to B.

Based on our experiments, accesses to B have a non-linear gating effect, in which they appear to 'transport' the Rowhammer effect of A over through B to C.

"Unlike TRRespass, which exploits the blind spots of manufacturer-dependent defenses, Half-Double is an intrinsic property of the underlying silicon substrate. This is likely an indication that the electrical coupling responsible for Rowhammer is a property of distance, which makes sense to me, the physics involved, effectively becoming stronger and longer ranged as cell geometries continue to shrink. Distances greater than two are conceivable.

"Google has been working with JEDEC, an independent semiconductor engineering trade organization, along with other industry partners, in search of possible solutions for the Rowhammer phenomenon. JEDEC has published two documents about DRAM and system-level mitigation techniques. We are disclosing this work because we believe that it significantly advances the understanding of the Rowhammer phenomenon, and that it will help both researchers and industry partners to work together to develop lasting solutions. The challenge is substantial, and the ramifications are industry-wide. We encourage all stakeholders (server, client, mobile, automotive, and IoT) to join the effort to develop a practical and effective solution that benefits all our users."

So everyone is worried about the possibility of what this would mean. But despite all the academic work that's been done, there have never been any reports of actual Rowhammer attacks in the wild. This is reminiscent of "Spectre" and "Meltdown." Right? But it might also be more relevant to the Y2K worry here, where despite the fact that the world did not end on Y2K, that may have been largely due to so much work going into making sure beforehand that it would not end. But in the case of all the various Rowhammer attacks, questions have been raised about the attack's true feasibility in real-world scenarios.

This brings us to the December 2024 presentation at Germany's 38th Chaos Communication Congress, during which a trio of academic researchers observed that the actual practical impact of these various RAM hammering attacks remains unknown and is still therefore largely theoretical. They noted that past academic research always used small, they considered them relatively microscopic, sample sizes.

They said: "The density of memory cells in modern DRAM is so high that disturbance errors, like the Rowhammer effect, have become quite frequent. An attacker can exploit Rowhammer to flip bits in inaccessible memory locations by reading the contents of nearby accessible memory rows. Since its discovery in 2014, we have seen a cat-and-mouse security game with a continuous stream of new attacks and new defenses. Now, in 2024, 10 years after Rowhammer was discovered, it's time to look back and reflect on the progress we've made and give an outlook on the future. Additionally, we will present an open-source framework to determine whether your system is vulnerable to Rowhammer.

"In 2014, researchers reported a new disturbance effect in modern DRAM that they called Rowhammer. The Rowhammer effect flips bits in inaccessible memory locations just by reading the content of nearby memory locations that are attacker-accessible. They trigger the Rowhammer effect by accessing memory locations at a high frequency, using memory accesses and flushes. The root problem behind Rowhammer is the continuous increase in cell density in modern DRAM. In early 2015, Seaborn and Dullien were the first to demonstrate the security impact of this new disturbance effect. In two different exploit variants, they demonstrated privilege escalation from the Google Chrome NaCl sandbox to native code execution, and from unprivileged native code execution to kernel privileges. Later, in 2015, Gruss et al. demonstrated that this effect can even be triggered from JavaScript, which they presented in their talk 'Rowhammer.js: Root privileges for web apps.'

"Now, in 2024, it is precisely 10 years after Rowhammer was observed. Thus, we believe it is time to look back and reflect on the progress we've made. We have seen a seemingly endless cat-and-mouse security game with a constant stream of new attacks and new defenses. We will discuss the milestone works throughout the last 10 years" - talking about the presentation they're about to give to the Chaos Congress - "including various mitigations (making certain instructions illegal, ECC, doubled-refresh rate, TRR [Targeted Row Refresh]) and how they have been bypassed.

"We show that new Rowhammer attacks pushed the boundaries further with each defense and challenge. While initial attacks required native code on Intel x86 with DDR3 memory, subsequent attacks have also been demonstrated on DDR4 and, more recently, on DDR5. Attacks have also been demonstrated on mobile ARM processors and AMD x86 desktop processors. Furthermore, instead of native code, attacks from sandboxed JavaScript or even remote attacks via network have been demonstrated, as well.

"Furthermore, we will discuss how the Rowhammer effect can be used to leak memory directly, as well as related effects such as RowPress. We will discuss these research results and show how they're connected. We will then talk about the lessons learned and derive areas around the Rowhammer effect that have not received sufficient attention so far. We will outline what the future of DRAM disturbance effects may look like, covering more recent effects and trends in computer systems and DRAM technology.

"Finally, an important aspect of our talk is that we invite everyone to contribute to solving one of the biggest unanswered questions about Rowhammer: What is the real-world prevalence of the Rowhammer effect? How many systems, in their current configurations, are vulnerable to Rowhammer? As large-scale studies with hundreds to thousands of systems are not easy to perform, such a study has not yet been performed. Therefore, we developed a new framework to check if your system is vulnerable to Rowhammer, incorporating the state-of-the-art Rowhammer techniques and tools. Thus we invite everyone to participate in this unique opportunity at the 38th Chaos Communication Congress to join forces and close this research gap."

The site, they called their overall work flippyr.am because it's flipping bits. So F-L-I-P-P-Y-R, flippyr.am. But the site has the dot between the R and the AM. So flippyr.am. You know, https://flippyr.am. That's where all of this lives. Anyone who's interested should go to flippyr.am, grab a copy of the open source test tool.

They say when you get there: "Welcome to our FLIPPYR.AM Study. We want to analyze the prevalence of Rowhammer in real-world systems. Everybody can participate in our study. The entire source code is open-source and available via GitHub. You can either build the ISO yourself or run the entire study using Docker. However, we highly recommend using the ISO image. And the ISO is just flippyr.am/hammeriso.iso."

They said: "Simply follow these steps: Download our ISO image and flash it to a USB thumb drive (see the following links for instructions for Windows, Mac, and Linux). Boot the system you want to test using the thumb drive you created before. Specify the time the experiment should run and confirm your participation in the study." And they said: "(When you do not want to participate in the study, you can still check if your system is vulnerable to Rowhammer without submitting any data.) Step 4, wait for the experiment to finish. Step 5, you'll get a brief overview of the results. Additionally, the raw results will be stored on the thumb drive for you to inspect afterwards. And 6, the results will be uploaded to our server, and you can access them using a URL shown at the end of the test (only if you confirmed to participate before)."

Okay. So first of all, you should know you are asked afterward if you want to do the upload. So there's nothing happening behind your back. None of your data will sneak

away. The default testing time is eight hours. So the idea being, you know, you run this overnight while you're not using your computer, and then it's done in the morning.

Leo: It's a probabilistic attack. It's not - it doesn't work every time.

Steve: Correct. Exactly. And so it requires some patience. And, you know, unfortunately they don't have anything cool like a running total on the screen of like Rowhammer strikes.

Leo: They should be talking to you. They could do this [crosstalk].

Steve: So you're not getting any results available on the way. It does take a while to get going. On my ZimaBoard, like I wasn't sure it was working because it went to, like, it has four stages, and it went to 100% on the first stage. Then it went to 55% on the second stage, where it sat for a long time. The first stage is fetching info, but that's not from the network, it's just from the system, apparently. Then retrieving addressing functions, that's stage two. And my ZimaBoard sat there for a long time. But I have also since then run it on one of - actually on a next-generation GRC server platform that I have not yet deployed. So, I mean, it's got, I don't know how many cores this thing has, 27 or something.

And, I mean, it is a screamer. It acted exactly the same way. It sat at 100% for a while, or it took a while to get to 100%. Then the second stage sat at 55 for a long time. Since I started it yesterday afternoon on the server and let it run until this morning, I let it run for 16 hours. I should have known nothing would show because this is a server platform with error-correcting, you know, it's got ECC RAM, server RAM, which is unusual. And it came out, it came back completely clean. But on the other hand it was nice to actually see that validated. So it will take some time. Once it finishes, you get a summary on the screen. It writes a long report in log files, in text, on another partition that it creates on your thumb drive, which you are able to look at.

And then this morning I got a big QR code that I took a picture of with my phone, and the phone also wanted to open it. And so I haven't had a chance to look at it. But there you get a detailed report from their server, which analyzes an incredible amount of information. I mean, these log files, I don't know how many hundreds of log files I had that it had written out. So anyway, for what it's worth, I'll be uploading, and I did, all of my results. And I would hope others would, too, to give them as large a cross-section. I think it would be interesting, if you have older machines, to see whether, you know, like old DDR3 or DDR4 machines, to see if they're actually vulnerable to Rowhammer attacks.

Leo: Now, they say Macintosh is - you can run this on a Mac?

Steve: Yeah.

Leo: Okay.

Steve: Yeah.

Leo: So it's not an x86...

Steve: Yeah, I don't have any non-x86 hardware here or I would have done that. But I imagine that it is multiplatform, Leo.

Leo: So anything with DDR3, 4, or 5.

Steve: Yup.

Leo: Is 5 immune?

Steve: No, 5's not immune. Attacks have surfaced for DDR5.

Leo: Okay.

Steve: Basically everything we have in the world now is still vulnerable to Rowhammer to some degree.

Leo: Yeah. Interesting.

Steve: And they said, I mean, this is dumb. They said: "As an incentive, the following two rewards can be won. When you upload a valid dataset, you'll receive a cryptographic token. This token is generated by hashing random data; and when you upload your dataset, we will save this token separately in our database. This means the token is not associated with your dataset. This ensures that you can participate in the raffle without linking the token to your dataset. Please make sure to bookmark or save the token."

Then they said: "The first 10 valid tokens they receive via email will get a flippyr.am T-shirt." I'm sure those are long since gone. And then "Everyone who sends us an email with a valid token will participate in a raffle and have your chance to win a 10 euro Amazon gift card."

Leo: Okay.

Steve: "The more tokens you send us, the higher your chances are." So token away. Anyway, they've got two releases of the tool so far, v1.0 and 1.0.1. They published the SHA256 hashes of both the ISOs, if you want to make sure that they weren't tampered with. Although I've never understood the logic of that because if someone was going to tamper with the ISO, they would just tamper with the posted SHA256 also.

Leo: Of course, yeah.

Steve: Anyway, fine. Anyway, at the bottom of the show notes I have a link to the Chaos Communication Congress presentation. It's a multilingual soundtrack, so it's probably

available in your language, if you want to listen to the whole presentation. And I hope our listeners will, you know, have some fun. Copy it to a thumb drive, run it on your machines overnight, see what you find out. Let me know via our Security Now! feedback because it would be fun just to share some of our listeners' results. And also submit your data to them. It's all anonymous, no information that you care about. I mean, you're booting from scratch, right, you know.

And they tell you, if you're worried about any of your mass storage devices, you know, disconnect them while you're running the test. And then the machine knows nothing about you, has no ability. But you can also look at the source code. And I'm sure these are good guys in any event. So a fun thing for our listeners to do.

Leo: Yeah, kind of interesting, yeah.

Steve: While you're waiting for Episode 1018.

Leo: And it runs for eight hours. That's the fixed amount of time. Or can it run for a different amount of time?

Steve: It defaults to eight. It's got hours and minutes in a little field, and you can change it. I changed it to 16 for my server...

Leo: Well, why not, yeah.

Steve: ...because I had 16 hours I was going to be away from it, so what the heck.

Leo: Right, yeah. And, I mean, honestly, it's conceivable that it wouldn't even get a hit in that amount of time. So, right? I mean, there's no, like I said, it's probabilistic. It's not...

Steve: It's going to be interesting to see what our listeners find. I did not get much satisfaction from the ZimaBoard. I think that its hardware, you know, it is sort of an embedded system.

Leo: Right.

Steve: So it's not a full PC. And a number of the tests that they had, the ZimaBoard did not qualify for.

Leo: Right, right.

Steve: So it'll just be interesting to have it run on more systems.

Leo: Very cool.