THE ANGLE OF THE DANGLE

Description: Why is updating your Windows laptop with last week's patches potentially much more important than usual? Copilot+'s Recall feature won't be released today; what happened? Was Recall recalled? What does Johns Hopkins' well-known cryptographer think about Apple's new Private Cloud Compute concept? How could the WGET command-line utility possibly have a CVSS 10.0 vulnerability? Or does it? What order did Google, Cloudflare, and Cisco recently receive from a Parisian court? And after a brief GRC email update and three pieces of closing-the-loop feedback from our listeners, we're going to examine exactly how Microsoft lost control of their code.microsoft.com subdomain and why the underlying problem is far bigger than them.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-979.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-979-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. You're going to get to watch us try out some new technology if you're watching the video. We're using Restream to produce this show today, which gives us some interesting new features. Steve's going to talk about Patch Tuesday. Holy cow. Microsoft patched one of the worst flaws I've ever heard. It gives Microsoft and all of us reason to think maybe Recall should be recalled. In fact, that's exactly what happened. Do you use WGET? Maybe you shouldn't. And then, finally, Steve's going to talk about how that kind of DNS poisoning that he talked about last week at code.microsoft.com, how that happens and how you need to configure DNS to prevent this. I'm talking to corporate IT folks. It's all coming up next on Security Now!. Stay here.

Leo Laporte: This is Security Now!, Episode 979, recorded June 18th, 2024: The Angle of the Dangle.

It's time for Security Now!, the show where we cover the latest news in security, privacy, security, healthiness, and sci-fi with that guy falling over because I'm going on so long with his intro, it's Steve Gibson of GRC.com. Hi, Steve.

Steve Gibson: Hello, Leo. I hope I sound the same to everyone, as good as I have for the last 20 years. We're on new technology today.

Leo: Yes, we are. We're testing out something called Restream. The sound, if you're an audio listener, which 90% of you are, should not be any different.

Steve: You sound just the same to me except the headphones are distracting because the cord's coming off of the wrong side. I don't know. I don't know.

Leo: Will it help if I do this?

Steve: Oh, it's much better. Now I recognize you. It's Leo.

Leo: Who is this guy with [crosstalk] headphones.

Steve: I recognize your voice, but, you know. Okay.

Leo: Actually, for years people have told me that I'm wearing my headphones backwards because this is supposed to be the left ear. So I was actually doing it - now I'm strangling myself. I was actually doing it - this is correct. Let's go back. Anyway, to explain what's going on, if you're watching video, it does look a little bit different, and that's because we're using a new program called Restream. And this is so that we can, sad to say, but a sad necessity, close the studio in a month or so and move to a local recording from my house. And this is how we're going to do it.

We won't be able to bring the TriCaster and all the beautiful Telos Axia hardware that we use for audio. But Restream is a really great solution that allows us to pretty much do the same thing that we've always done. We have - Benito's here. He's our TD remotely. So he's at his house. He can - everybody can work from home.

Steve: What?

Leo: Yeah.

Steve: Cool.

Leo: And so he's going to watch the show. Kevin King's probably going to do it, too. Depends who ends up producing it. But the producer of the show will watch the show, can switch. But I said, no, I want the capability to do things like this. And this. Woo, it's fun. Steve was flipping himself back and forwards like this earlier. So in a way this is kind of a cool platform because it gives me some interesting capabilities, one of which I really think we're going to enjoy, which is I can pull up, from chat, people talking. So if you're in the chatroom, like Emanuel is, you can chat with us. Now, we can see from the little icon that he's watching on YouTube. The other thing that happens with this is we're able to stream onto Twitch, YouTube, and other platforms, as well.

Steve: Did we lose Discord?

Leo: We don't stream video into Discord. But you know what, video on Discord was always kind of bad.

Steve: I would keep losing the audio when I was trying to watch.

Leo: Yeah, it's not good. So we're going to push people - and we can actually do X, Instagram, and other platforms, as well. So eventually this will give us the capability to be in more places. And, you know, Emanuel's chat was from the YouTube chat. So chatters from everywhere, watching on any stream, will actually be able to converse with us. And if there are good comments, instead of me reading them, I can pull them up. So that's pretty nice. So the Discord chat is still there. The video - that's Patrick, our engineer. The video is not. It was meh, as Patrick said. So I think this is going to be, once my fingers become trained...

Steve: Tuned, yes.

Leo: Yeah, because it is, it's a lot of stuff to learn for me. But I think there's a lot of benefit to this. One big dysfunctional family, as DCrash says. Everybody's included, Lou. So this is going to be fun. Now, Steve, yes, the real question is what's coming up on the show today?

Steve: So we've got a bunch of interesting things to talk about. We're going to learn why updating your Windows laptop with last week's patches is potentially more important than it has been for a long time. Also, today's June 18th. Copilot+'s Recall feature...

Leo: That's right, yeah.

Steve: ...will not be released today.

Leo: Nope.

Steve: What happened?

Leo: Victory.

Steve: Was Recall recalled? I think so.

Leo: I can't recall.

Steve: Also, what does Johns Hopkins well-known cryptographer think about Apple's new Private Cloud Compute concept? How could the WGET command-line utility possibly have a CVSS vulnerability rating of 10.0, which we know is reserved for, you know, end-of-the-world events? Or does it? What order did Google, Cloudflare, and Cisco recently receive from a Parisian court? And after a brief GRC email update and three pieces of closing-the-loop feedback from our listeners, we're going to examine exactly how Microsoft lost control of their code.microsoft.com subdomain.

Leo: Oh, okay.

Steve: And why the underlying problem is far bigger than them. Thus today's podcast title, "The Angle of the Dangle."

Leo: Ah.

Steve: For Security Now! #979, closing in on that magic 999. Will we be able to go to four digits? That's the question. What will break? It's our own Y2K. We'll find out.

Leo: Is 999 going to break something?

Steve: Yeah. My whole system actually, as I said in the beginning, the reason the stopping at 999 ever was brought up was that my system will collapse if I put in four digits because I thought, you know, back in, what, 1923 when we began this, I thought, we're never going to be doing this in 20 years.

Leo: Yikes.

Steve: Maybe we're going to run out of steam after the 15th podcast. But here we're going.

Leo: Well, we'd have to face it. But you know what, I think there's probably more than a few people who say, whatever it takes in 21 episodes, we're glad that there's going to be an Episode 1000.

Steve: Yes. And I will spend some time to figure out - actually, the problem was that I was redirecting clicks at GRC.com through Podtrac in order to give TWiT credit for...

Leo: Oh, but you don't have to do that anymore. We don't do that anymore.

Steve: Well, so then there's not going to be any problem. I'll just pull the plug on the Podtrac TWiT redirector code, and we're good to go.

Leo: Okay.

Steve: Yeah.

Leo: All right. We're going to pause, and Leo's going to push a bunch of buttons including Shift Nancy, which apparently...

Steve: If I'm still here after that, then we'll have a podcast.

Leo: Shift Nancy did the job. And now I'm going to click this link, and welcome...

Steve: Thought it was Nora.

Leo: ...Steve Gibson in his continuing quest to make us all safer online. It's all yours, Steve.

Steve: So to that end we actually do have, we're going to end up with some interesting takeaway. Our Picture of the Week is sort of a placeholder. You know, it's not one of our crazy fences out in the middle of nowhere with a bunch of sheep standing behind the fence, not willing to go around for some reason. That one was a real puzzler. This is a diagram that demonstrates the process of CNAME, DNS record-based subdomain takeover. And I gave this...

Leo: Oh. Well.

Steve: Yes, yeah. And it's got lots of pretty - it looks like something you would get, you would use your box of Crayola crayons to color in. I gave it the title, "Fortunately, as we'll see today, the 'Subdomain Takeover' problem is much less confusing than this diagram!" because, I mean, I understood the problem immediately when I realized that's what had happened at Microsoft. I think maybe, Leo, every one of our listeners said, "I know what happened. I'm going to tell Steve." So, boy, does the email system work. But anyway, I had to sort of figure out the diagram after already knowing what the problem was. So just for what it's worth, to anybody who, like, gets lost in these arrows, don't worry. By the end of the podcast, you'll understand what's going on.

Leo: Good. Because it's not obvious.

Steve: No, it's not. The diagram does not help much. I need to begin this week by making 100% certain that everyone listening is aware of a flaw that was patched during last Tuesday's Windows patch fest. It's CVE-2024-30078. And the only reason that it only has a CVSS score of 8.8, rather than 11, on a scale that maxes out at 10, is that an attacker needs to be within WiFi radio range of any Windows machine.

Leo: Oh.

Steve: Uh-huh, any Windows machine with WiFi enabled that has not been updated with last Tuesday's patches.

Leo: Oh, that's terrible.

Steve: It's unbelievably bad.

Leo: Because one network might have multiple Windows machines, and one that's close to the window.

Steve: Yes.

Leo: Right?

Steve: Being, you know, that it's Windows. And, yeah. Or somebody has a laptop and, you know, for whatever reason they've learned, for example, that, well, you may not want to apply those patches immediately. Let them kind of sit for a while to see if Microsoft made a mistake, and apply them later. Okay. In this case, we have the worst of all possible problems.

Last Tuesday eliminated a fortunately very rare remote wireless radio takeover of any Windows machine which is using its native WiFi protocol stack.

Leo: Geez.

Steve: This affects all versions of Windows ever, so that means that any machine that was not updated, as I said, for whatever reason last Tuesday, is vulnerable today. Microsoft is only saying "all supported versions of Windows," but that's what they always say. And that typically means all earlier versions, too, but they don't want to say that because they've decided they don't want to support those anymore. So good luck to you. And the fact that this appears to be Windows-wide suggests that the flaw was in a core component that they have not been messing with for the last, you know, 15 years. Because, as we know, mostly Windows is just new levels of, you know, new layers of candy coating on top of the core, which they're afraid to touch. Which thank god, you know, leave some of it alone, at least.

So to provide some additional color and perspective, I'll share what Forbes' cybersecurity guy wrote under their headline "New Wi-Fi Takeover Attack - All Windows Users Warned To Update Now." The guy wrote: "Microsoft has confirmed a new and quite alarming WiFi vulnerability in Windows, which has been rated 8.8 out of 10 in terms of severity using the Common Vulnerability Scoring System. The vulnerability, assigned as CVE-2024-30078, does not require an attacker to have physical access to the targeted computer." Exactly as you said, Leo, just be standing outside the building. "Although physical proximity" - meaning you cannot do it from Russia. In fact, that's why it's not a 10.0. If you could do it from Russia it would be a 10.0. But you've got to be nearby, you know, WiFi range.

"Exploiting this vulnerability," he writes, "can allow an unauthenticated attacker to gain remote code execution on the impacted device. What's perhaps most concerning, though, is that this WiFi driver security flaw affects all supported versions of the Windows operating system. Microsoft has confirmed that with no special access conditions or extenuating circumstances, apart from the proximity requirement, an attacker could 'expect repeatable success against the vulnerable component.' Microsoft also warns that an attacker requires no authentication as a user on that machine before exploiting this vulnerability, nor any access to settings or files on the victim's machine before carrying out the attack. Further, the user of the targeted device does not need to interact at all. There's no link to click, no image to load, no file to execute.

"Jason Kikta, chief information security officer at Automox, said that, given its nature, 'this vulnerability poses a significant risk in endpoint-dense environments including hotels, trade shows, or anywhere else numerous devices connect to WiFi networks.' In these kinds of environments, it would be all too easy for an attacker to target users

without raising any red flags. To protect against this vulnerability, it's recommended that you apply the latest patches as soon as possible.

"Assuming, that is, you are using a version of Windows that still receives security updates. 'Anyone using an end-of-life version of Windows without an extended service contract is recommended to update to a supported version as soon as possible,' Kikta said. 'If patching immediately isn't feasible, you must use endpoint detection to monitor for suspicious activity related to this vulnerability. Due to its unique nature, it's unlikely to be visible to network-level detection methods.'" Meaning, you know, it's down in the kernel, you know, deep stack, before this even gets up to the network interface level at a higher level. "He says: 'The risk of running outdated software cannot be overstated.'"

The article then concludes: "In case you need any further incentive to get patching as soon as possible, Kikta said: 'This close access vector threat potentially bypasses network-based detections and mitigations. It circumvents most threat modeling, so this is an immediate patch priority for me,' he said. 'Most security experts agree that publicly available exploitation tools will be available before long, so the window of opportunity to patch before the attacks start is getting smaller every day.'"

And I should note that GitHub does indeed already have a vulnerability detection and command execution script posted. So that has happened. Not malicious, but it's easy to take that script and make - actually they did have - they had it commented out where you're able to provide the command. So technically not malicious, but it doesn't even take a script kiddie in order to make this do something bad.

Leo: Wow.

Steve: Microsoft, in their own tracking of this CVE, enumerates the following characteristics, which are bracing. They said under "Attack Vector," they said "Adjacent." And they wrote: "The vulnerable component is bound to the network stack, but the attack is limited at the protocol level to a logically adjacent topology. This can mean an attack must be launched from the same shared physical, i.e., Bluetooth or IEEE 802.11, you know, which is WiFi; or logical, local IP subnet network; or from a secure or otherwise limited administrative domain."

Under "Attack Complexity," this is not what you want to hear, "Low Complexity attack." They said: "Specialized access conditions or extenuating circumstances do not exist. An attacker can expect repeatable success against a vulnerable component." But privilege is required.

Leo: Wow, this is bad.

Steve: Leo, this is as bad as it gets. This is a shocking flaw for Windows to have.

Leo: CBits wants to know if the firewall stops it.

Steve: No.

Leo: Oh, my god.

Steve: The firewall is at the border, and this is inside the boundary that the firewall protects.

Leo: Unbelievable.

Steve: For "Privileges Required: None. The attacker is unauthorized prior to attack, and therefore does not require any access to settings or files to carry out the attack. User Interaction: None. The vulnerable system can be exploited without any interaction from any user. Confidentiality," they said, "High Threat. There is total loss of confidentiality." And this is where the favorite comment I saw on the Internet was, "But don't worry, your Recall data is encrypted." It's like, uh-huh, right.

So, I mean, this is exactly the kind of problem that Microsoft ignores when they say, oh, storing all the history of your computer, not a problem. We've got you. We've got your back. And here's a flaw in the WiFi protocol stack that probably dates back 15 years. This is, you know, I think 7 is out of extended service now, but 8 is still in it. And 8 is vulnerable. So we're talking this has been around for a long time. And we don't know who already knows about it. We just know that it finally came to light.

And for Integrity they said: "High Threat. There is a total loss of integrity, or a total loss of protection." This is Microsoft saying this of their own flaw. "For example, the attacker is able to modify any/all files protected by the impacted component." And then they wrapped it up with two FAQ questions. They said: "Question: According to the CVSS metric, the attack vector is adjacent. What does that mean for this vulnerability? Answer: Exploiting this vulnerability requires an attacker to be within proximity of the target system to send and receive radio transmissions." And the second question: "How could an attacker exploit the vulnerability? Answer: An unauthenticated attacker could send a malicious networking packet to an adjacent system that is employing a WiFi networking adapter, which could enable remote code execution."

So, you know, the appearance, as I said, the appearance of this vulnerability provides a perfect case-in-point demonstration of why the presence of Recall running in Windows machines represents the threat that it does. I have no doubt that Microsoft's heart - their heart, not their brain - is in the right place. They're not an evil empire. But they're now attempting to add an extremely powerful and high-risk feature to an old, creaky, and bug-ridden operating system. That's the only way you can characterize Windows, an old, creaky, bug-ridden operating system. Again, they keep adding. They, you know, change the UI and make the desktop look different. But we all see dialogs that we remember from Window 95, so it's still in there. Well, okay, Windows 2000.

Anyway, this vulnerability demonstrates what we all intuitively feel, which is that Windows is not up to the task of protecting anything as valuable as everything that our machine has ever displayed on its screen. You know?

Leo: The scenario is very simple. You've got Recall running. Grandma has Recall running on her home network. She's got WiFi running. Some guy sits out on the curb, joins her computer that's in the window - she doesn't even have to be on that one - can then easily get into the other computers and access the data.

Steve: Yeah.

Leo: I mean, it's a very - this is why you hated the idea of Recall in the beginning.

Steve: Yes. Yes. Exactly. You know, it's just they don't, you know, Microsoft has come up with, I would argue, a killer idea, the idea of storing all of the history and then, probably in the future, creating your own personal AI expert that knows everything you've done. That, I mean, as I said when we first discussed it, that is transformative. It could change, it would, will change people's relationship with computing. I mean, I think it's really huge. The problem is they haven't been spending any time on the security of their system. They keep giving us ridiculous crap features that nobody wants. And we have, what, 53 flaws I think this month, 100, more than 100 last month.

Leo: Which is low. 53 is low. That's the thing. It's a small number, relatively.

Steve: And then here this comes along, which is in every version of Windows. It's been in there for who knows how long. So I just, you know, it just, I mean...

Leo: Amazing.

Steve: I'm sure, as I said, somebody, the smart people at Microsoft are thinking, wow, wouldn't it be cool if we actually had an operating system that was safe to put this kind of capability into. They don't. And they just can't get one by saying they do. You know, we have a history of Microsoft declaring each version of Windows is more secure than the last one. Meanwhile, this has been in there the whole time. And who knows what else? How many flaws have they got queued up for July?

Leo: So, I mean, and just to be clear, this Recall's not a crap feature. It's a feature people would want, and really cool. It's just you're putting it on a platform that isn't secure.

Steve: Right. It is a rickety, insecure platform. And that's fine if you want to run Excel. Okay, Leo, I know you've messed with security certificates in the past. Have you ever looked at a certificate just to sort of make sure that it was correct?

Leo: No.

Steve: You know, we all know what they look like; right?

Leo: I mean, I've used Let's Encrypt. I have Let's Encrypt certificates on my NAS and on various places. I never look at them.

Steve: I would imagine, well, they're just text. It's a little text file, a 2K or 4K text file. And if you look at it, you can see that it's a nicely, you know, it's a Base64 encoded blob. If that appears on your screen, and Recall snaps it, you've just lost privacy of your private key.

Leo: I never thought of that. So the information that somebody would need, the private key is in plaintext in that...

Steve: Yes, yes. Certificates are viewable as text.

Leo: The private key. So in public/private key crypto, the public key can be given to everybody, and is, and that's why it works. But you keep that private key close to your chest because if somebody got it, they can impersonate you.

Steve: And if you ever open it on your computer and look at it, Recall snaps it.

Leo: I never thought of that. Of course they could have a switch that says, oh, yeah, if there's a certificate, don't look at it. Because they have some things don't look at it. But then, and then...

Steve: And then, and then, and then, exactly. Yeah.

Leo: Because we do stuff, private stuff on our computers. Duh.

Steve: Yeah. You know, and we want to be able to. I've been thinking about this a lot. And I'm thinking, okay, if you were conscious of the fact that your computer was spying on you because you want it to, I mean, like, with your permission it's trying to collect all of this. But then this does, for anyone who's privacy-minded, they've got to be constantly turning it off before, not after.

Leo: Right.

Steve: Although I guess you are able to go back and kill, like, some period of time.

Leo: Oh, good luck editing stuff.

Steve: So, oh, crap, I forgot to turn off Recall.

Leo: Yeah, yeah.

Steve: Again, you've got to remember. So if you want this, and you want privacy, and you're serious about it, then you're constantly toggling this thing on and off so that it doesn't capture things that it shouldn't capture that you don't want to have your expert know. Otherwise, Russia gets into your computer and says, uh, show me all of the private keys that have been on the screen. Ping, there they are. So, yes, super search.

Leo: Wow.

Steve: Speaking of Recall, first they switched Recall to opt-in, which we covered last week. That was a welcome move.

Leo: Yes.

Steve: But as we know, I was still quite worried that Microsoft would nevertheless be promoting the heck out of it. And why would Microsoft ever mention the very real danger inherent in having any machine storing its entire history? You know, they're not going to say, oh, well, we want you to do this, but danger, Will Robinson. No, they're going to just - they're not going to tell you what could go wrong. So the latest good news is that Recall has now been completely removed from today's June 18th production release of Windows Copilot+. It will initially only be available to users participating in the Windows Insider Program.

Last week, I shared Microsoft's blog posting where, amid all of their mumbo jumbo about how they're putting security first, they're explaining, you know, that they're going to be switching Recall to opt-in and only decrypting, you know, on the fly after you've given them a blood sample and chapped your heels three times. In an update to that blog posting last Thursday, they added: "Update: June 13th, 2024: Today, we are communicating an additional update on the Recall (preview) feature for Copilot+ PCs. Recall will now shift from a preview experience broadly available for Copilot+ PCs on June 18, 2024, to a preview available first only in the Windows Insider Program (WIP) in the coming weeks. Following receiving feedback on Recall from our Windows Insider Community, as we typically do, we plan to make Recall (preview) available for all Copilot+ PCs coming soon."

So again, yay, they backed off. They said: "We're adjusting the release model for Recall to leverage the expertise of the Windows Insider community to ensure the experience meets our high standards for quality and security. This decision is rooted in our commitment to providing a trusted, secure, and robust experience for all customers and to seek additional feedback prior to making the feature available to all Copilot+ PC users. So nobody gets Recall - yay - until the insiders have played with it, people have experimented with it, and we've learned collectively more about it."

So this is terrific news. And I'm sure Microsoft will get there eventually, if it wishes to. And I'm quite certain that it wishes to. As I said, I believe Microsoft has more in store for a machine's entire usage history than just scrolling back through time, since the actual delivered security of anything can only be judged retrospectively. That means that only time will tell whether they've been able to sufficiently protect any machine's entire usage history. What they were on the brink of was obviously half-baked, if that.

So it appears that they allowed their enthusiasm for this new capability to get the better of them. And as we saw from the reactions of the entire industry, this sort of immature exuberance does not inspire confidence. The main point I wanted to make is to note that we've never seen anything like Recall before. It is not just a change in degree. This represents a change in kind, and it presents a security challenge of an entirely different scale. And that's just something Microsoft glossed over because, you know, they wanted to.

Leo: Yup.

Steve: And, you know, maybe they're reading their own press. I don't know.

Leo: I think they, you know, I mean, in their defense, they heard you and all the other security researchers howling. And they did respond to it. I mean...

Steve: No, I'm 100% in agreement. They realized this thing was going to come out and just get panned by, you know...

Leo: Problems.

Steve: ...by, like, everyone.

Leo: Yeah, yeah. All right. You want to take a break here, my friend?

Steve: I do indeed.

Leo: I have my finger hovering over the button.

Steve: Then we're going to talk about Matthew Green and what he thinks about Apple's private...

Leo: You know, it's funny, because you're going to talk about Apple in the next one. And I'm thinking it's very easy for me, an Apple user and not such a big fan of Windows, to go, yeah, Microsoft. But now, when the shoe's on the other foot, I'm, whoop, let's see. I may not be so sanguine about the whole thing. We'll be listening. Just a moment. But first, I think, whoops, I've got to do the right thing, Shift Nancy, and tell you about - I'm still learning the keys here. Why are you laughing, Steve? What are you laughing at?

Steve: I'm just smiling.

Leo: He's okay. Steve Gibson.

Steve: So to remind everyone of Matthew's pedigree, this is Matthew Green, he's an American cryptographer and security technologist, Associate Professor of Computer Science at the Johns Hopkins Information Security Institute. And we're fond of quoting Matthew.

Leo: He's great.

Steve: He's often outspoken about security issues. And super trustworthy.

Leo: Super trustworthy. Super trustworthy, yes.

Steve: So in a series of, I think it was 21 individual sequential postings under his Mastodon account, he outlined his feelings about what Apple described and about the nature of the challenge that they're undertaking with their private cloud compute

concept. And I'm going to share this because it was great, and then we'll also talk about it.

So he said: "So Apple has introduced a new system called Private Cloud Compute that allows your phone to offload complex," he says, "typically AI tasks to specialized secure devices in the cloud." He said: "I'm still trying to work out what I think about this. So here's a thread. Apple, unlike most other mobile providers, has traditionally done a lot of processing on-device. For example, all of the machine learning and OCR text recognition on photos is done right on your device.

"The problem is that while modern phone 'neural'" - he has in quotes - "hardware is improving, it's not improving fast enough to take advantage of all the crazy features Silicon Valley wants from modern AI, including generative AI and its ilk. This fundamentally requires servers. But if you send your tasks out to servers in the cloud, this means sending incredibly private data off your phone and out over the Internet. That exposes you to spying, hacking, and the data-hungry business model of Silicon Valley.

"The solution Apple has come up with is to try to build secure and trustworthy hardware in their own data centers. Your phone can then 'outsource' heavy tasks to this hardware. Seems easy; right? Well, here's the blog post." And then he provided a link to Apple's own disclosure about their private cloud compute.

And he said: "TL;DR: It is not easy. Building trustworthy computers is literally the hardest problem in computer security. Honestly," he wrote, "it's almost the only problem in computer security. But while it remains a challenging problem, we've made a lot of advances. Apple is using almost all of them." He said: "The first thing Apple is doing is using all of the advances they've made in building secure phones and PCs in their new servers. This involves using Secure Boot and a Secure Enclave Processor (SEP) to hold keys. They've presumably turned on all the processor security features. Then they're throwing all kinds of processes at the server hardware to make sure the hardware is not tampered with." He says: "I can't tell if this prevents hardware attacks, but it seems like a start."

Okay. And then Matthew includes a screen shot from Apple's posting, which explains. This is Apple: "Private Cloud Compute hardware" - get a load of this. Wow. I mean, this, I'm a bit of a fanboy for this - "hardware security starts at manufacturing, where we inventory and perform high-resolution imaging of the components of the Private Cloud Compute node before each server is sealed and its tamper switch is activated. When they arrive in the data center, we perform extensive revalidation before the servers are allowed to be provisioned for PCC (Private Cloud Compute). The process involves multiple Apple teams that cross-check data from independent sources, and the process is further monitored by a third-party observer not affiliated with Apple. At the end, a certificate is issued for keys rooted in the Secure Enclave UID for each PCC node. The user's device will not send data to any PCC nodes if it cannot validate their certificates."

I'm just, you know, nobody has ever done anything like this before. And as I said, I need to confess that I'm a bit of a fanboy for the idea that Apple is performing high-resolution imaging of each node's components in order to detect anything that might have been done to the server between its design and through its manufacturing. That's very cool. And just the fact that it's now widely known that this is being done, likely serves as a deterrent to prevent anyone from even trying to mess with them.

Matt continued. He said: "They also use a bunch of protections to ensure that software is legitimate. One is that the software is 'stateless' and allegedly does not retain any information between user requests. To help ensure this," he writes, "each server/node reboot re-keys and wipes all storage." So the idea is it's like Apple has provided a large and growing collection of remote servers, and an individual iPhone user is able to be

connected to one of these, exchange keys, independently validate that specific connected node's security, establish a secure tunnel, send the data to this big computation engine, have it do whatever it needs to do on behalf of its user, return the results, and then it shuts down, reboots, wipes memory.

Leo: Wow.

Steve: And then comes back up again so that it's fresh and clean for the next user.

Leo: For each request?

Steve: Yes.

Leo: For every request?

Steve: Yes.

Leo: It reboots for every request?

Steve: It cleans itself out, rekeys, wipes all storage.

Leo: That seems like a lot.

Steve: Yup. That's Apple.

Leo: But that's what you have to do; right?

Steve: Yeah.

Leo: But, I mean, there's going to be millions of people using this in a second. They're going to have a million machines rebooting every second?

Steve: Maybe it boots fast, or they have a partition.

Leo: They're running in RAM or something. I don't know. That's wild.

Steve: Yeah. Again, he quotes Apple's announcement, saying - this is Apple: "We designed Private Cloud Compute to make several guarantees about the way it handles user data." Three of them. "First, a user's device sends data to PCC for the sole, exclusive purpose of fulfilling the user's inference request. PCC uses that data only to perform the operations requested by the user. Two, user data stays on the PCC nodes

that are processing the request only until the response is returned. PCC deletes the user's data after fulfilling the request, and no user data is retained in any form after the response is returned. And then, three, user data is never available to Apple, even to staff with administrative access to the production service or hardware." I mean, so they've literally created a system that they themselves cannot penetrate.

Matt continues: "A second protection is that the operating system can 'attest' to the software image it's running. Specifically, it signs a hash of the software and shares this with every phone and client. If you trust this infrastructure, you'll know it's running a specific piece of software. Of course, knowing that the phone is running a specific piece of software doesn't help if you don't trust the software. So Apple plans to put each binary image into a 'transparency log' and publish the software. But here's a sticky point: not with the full source code." And again, Apple is still a private company; right? You know, they're not Linux.

So Matt quoted Apple, saying: "Our commitment to verifiable transparency includes" - and here we have four short points. "Publishing the measurements of all code running on PCC in an append-only and cryptographically tamper-proof transparency log. Making the log and associated binary software images publicly available for inspection and validation by privacy and security experts. Publishing and maintaining an official set of tools for researchers analyzing PCC node software." So they're going to create and provide tools that allow researchers to examine everything that they've done. And fourth: "Rewarding important research findings through the Apple Security Bounty program."

Matt says: "Security researchers will get 'some code' and a VM they can use to run the software. They'll then have to reverse-engineer the binaries to see if they're doing unexpected things." He says: "It's a little suboptimal." But again, you know, there's a limit to what you can ask Apple to give. So he says: "When your phone wants to outsource a task, it will contact Apple and obtain a list of servers, nodes, and their keys. It will then encrypt its request to all servers, and one will process it. They're even using fancy anonymous credentials and a third-party relay to hide your IP from themselves." So they're even masking the IPs of the people using this incredible resource in their own data centers.

Quoting Apple about this, they wrote: "Target diffusion starts with the request metadata, which leaves out any personally identifiable information about the source device or user, and includes only limited contextual data about the request that's required to enable routing to the appropriate model. This metadata is the only part of the user's request that is available to load balancers and other data center components running outside of the PCC trust boundary." In other words, the metadata's not encrypted in the tunnel, it's just used to get the tunnel endpoint connected within the Compute Center.

They said: "The metadata also includes a single-use credential, based on RSA Blind Signatures, to authorize valid requests without tying them to a specific user." Again, you can't even figure out who's doing the asking. "Additionally, PCC requests go through an OHTTP relay operated by a third party" - that's an anonymizing HTTP relay "which hides the device's source IP address before the request ever reaches the PCC infrastructure." Sort of like a mini TOR. "This prevents an attacker from using an IP address to identify requests or associate them with an individual." Again, Apple has gone so far beyond what anyone has ever done. "It also means," they write, "that an attacker would have to compromise both the third-party relay and our load balancer to steer traffic based on the source IP address."

So again, Matt says: "Okay. There are probably half a dozen more technical details in the blog post. It's a very thoughtful design." He said: "Indeed, if you gave an excellent team a huge pile of money and told them to build the best 'private' cloud in the world, it would probably look like this." He says: "But now the tough questions. Is it a good idea? And is

it as secure as what Apple does today? Most importantly, can users opt-out entirely from this feature?" He said: "I admit that, as I learned about this feature, it made me kind of sad. The thought that was going through my head, is this going to be too much of a temptation? Once you can 'safely'" - and he has that in air quotes. "Once you can 'safely' outsource tasks to the cloud, why bother doing them locally?" And Leo, I think the answer is what you just said. There's a limit to what this, like how much processing Apple could possibly provide in their cloud.

Leo: Oh, yeah. Everybody wants to do this on device eventually, not just for privacy but for economy.

Steve: Right, right. He says: "Once you can safely outsource tasks to the cloud, why bother doing them locally? Outsource everything." And the answer is, exactly as you said, you know, it's way better if your distributed computing is in everybody's hands instead of monster servers in the cloud. He said: "As best I can tell, Apple does not have any explicit plans to announce when your data is going off-device to Private Compute." And I notice that his saying that does not feel Apple-esque to me. It feels like Apple will provide these controls. We just haven't seen any of it yet. But they haven't said, you know, they haven't talked about that.

Matt said: "You won't opt into this. You won't necessarily even be told it's happening. It will just happen magically." He says: "I don't love that part." Now, maybe Matt knows something we don't. Or maybe we haven't seen that yet from Apple. He said: "Finally, there are so many invisible sharp edges that could exist in a system like this. Hardware flaws. Issues with the cryptographic attestation framework. Clever software exploits. Many of these will be hard for security researchers to detect. That worries me, too."

And that's an interesting point. We've talked about how, because Apple's smartphone handset technology is so tightly locked down in order to keep bad guys out, unfortunately it also keeps good guys from being able to see what they need to see in order to know what's going on. Remember that Kaspersky, who are among the best people there are, they had malware in their phones that they couldn't detect until they detected some strange network activity that was being driven by the malware, and that allowed them to begin to pursue what was going on. But, you know, they can't see inside Apple's iPhones any more than the bad guys can.

Leo: And that's been a complaint of security researchers forever.

Steve: Right.

Leo: To which Apple, by the way, has responded by saying, okay, we're going to give select researchers access. I think they need to have kind of a valve, an escape valve so that researchers, legitimate researchers can look in.

Steve: Yeah.

Leo: But I understand why they don't want to do that. They don't want to [crosstalk].

Steve: And it - right. And Matt's sort of contradicting himself because he, I mean, he's just saying, okay, you know, he's playing devil's advocate because he just told us that Apple will also be making a lot of this as open as they reasonably can.

Leo: Yeah.

Steve: I mean, providing virtual machines for...

Leo: That's mindboggling.

Steve: For people to poke at. Wow.

Leo: That's really great. I mean, that's what they need to do.

Steve: So he said: "Wrapping up on a more positive note, it's worth keeping in mind that sometimes the perfect is the enemy of the really good."

Leo: Yeah.

Steve: "In practice, the alternative to on-device is ship private data to OpenAI or someplace sketchier, where who knows what might happen to it."

Leo: Right.

Steve: He says: "And of course keep in mind that super-spies are not your biggest adversary. For many people your biggest adversary is the company who sold you your device and software." He says: "This PCC system represents a real commitment by Apple not to 'peek' at your data. That's a big deal." And his final tweet was, or whatever you call it on Mastodon...

Leo: Toot. We call them toots.

Steve: Toot, well, okay. His final toot. Bean eater. "In any case, this is the world we're moving to," he says. "Your phone might seem to be in your pocket, but a part of it lives 2,000 miles away in a data center. As security folks, we probably need to get used to that fact and do the best we can to make sure all parts are secure."

And I think Matthew's take is exactly right. The design of this system is what you would get if a bunch of very good engineers and cryptologists were to deliberately design a system that was meant to transiently extend an individual smartphone's local computing into the cloud for the purpose of performing some very heavy lifting. It takes advantage of everything we know about how to do this safely and securely. It will enable Apple's devices to do things no other mobile devices can do.

But I have a concern that Matt did not raise, which is that, because Apple has made this transparent to their users, no one will be able to appreciate the lengths Apple has gone to, to securely offer this capability. The listeners of this podcast understand that Apple is visually inspecting the motherboards of their servers prior to deployment because, for example, we've covered the worries over tiny chips being added to server hardware or Cisco routers when they're intercepted during transshipping. Even though that's way out there, it's a factor Apple has preemptively considered. Who else is going to go to these extremes?

It's not that I'm worried about Apple being underappreciated. It's that I can easily see "me, too" vendors popping up and offering their own outwardly similar capabilities that APPEAR to be the same...

Leo: Ah, yes.

Steve: ...while providing none of the same true protections. They'll be able to say: "We're doing the same thing Apple is doing," thus riding on Apple's coattails while providing far less true security for their users, at a far lower cost to themselves. The concern is that Apple is legitimizing and popularizing the idea of exporting what could be an extremely personal mobile device data blob to the cloud for cloud-based processing. Other vendors are going to do the same. But users of those lookalike services will be placing their users' data at far greater risk than Apple. And who would ever know?

Leo: Well, I'll tell you who would know. People who listen to this show would know; right? And I think what Apple counts on, you're right, the normal people will not, you know, know this. But what Apple counts on is that the people who do understand it who listen to this show will then kind of spread the word. And when their less sophisticated friends and family say, well, is this trustworthy, they'll say, oh, yeah, you should see what Apple has done. They don't need to go into the details. That's why you should listen to the show.

Steve: He, Leo, we have a big improvement with this new technology. Oh...

Leo: Your mug is still giant, Steve.

Steve: It wasn't darkening my screen. But it is darkening my screen.

Leo: Now, Leo's going to push some buttons. He's going to click a button here, click a button there. He's going to press Shift November, and now Steve can...

Steve: Now I know my name.

Leo: Now Steve knows his name...

Steve: Now I know my name.

Leo: ...and we'll continue with Security Now!. Thank you, Steve.

Steve: Okay. So there's buzz in the industry today - today, Tuesday, June 18th - of a recently discovered flaw in the widely used WGET command-line utility.

Leo: Oh, no.

Steve: Yeah.

Leo: I use this. I use it all the time.

Steve: Actually, it's the way I download the podcast audio every week to recompress it for Elaine.

Leo: [Muttering]

Steve: Some outlets are claiming that this flaw carries an attention-getting CVSS score of 10.0. But anyone reading that, anyone who's been listening to this podcast for long should immediately be skeptical. As we've seen, 10.0 scores are pretty much reserved for "end of the world as we've known it" flaws, and it's hard to see how you can have an end-of-the-world flaw that's not remotely exploitable, and probably also wormable without any user interaction at the receiving end.

But WGET is not a server or a service. It's just a convenient command-line tool used to retrieve files. As I said, I use it every week to grab this podcast's audio for recompression before I post it for Elaine. So how any flaw in any command-line tool that's not publicly exposing a vulnerable service to the Internet could rate a 10.0 is beyond me. I mean, we've seen bad ones that, like, get the 9.8. And it's like, oh, it got really close, but no. And they're bad.

But okay. So I did some digging, and it's true that there is a problem with WGET up through version 1.24.5. The problem surrounds incorrect parsing of semicolons appearing in the "userinfo" portion of the URL that's passed to WGET. Okay, now, for those who've been around for a while, you may remember that URLs are technically able to carry a username and password, which appears before the hostname. So rather than, for example, actually the example I'll use is example. Rather than https://example.com, you could have https://username: password@example.com. And it's that username:password and the @ sign which is the userinfo portion of a URL. It's sort of been deprecated. There's still some use for it, but you have to be careful because a username and password in a URL is now considered bad form. So, you know, use with caution. So the concern is that mishandled semicolons in that portion of a URL might lead WGET's parser to confuse the userinfo with the hostname which follows it in some way.

I located the dialogue with the guy who patched this flaw a few weeks ago. He wrote: "I just pushed a fix for the issue. Indeed, the URL parser implementation of WGET 1.x is based on RFC 2396, a standard from 1998. But even by this outdated standard," he wrote, "the implementation of the userinfo parsing was not correct. It hopefully is correct now. Anyway, nobody is going to lift the whole URL parsing of WGET 1.0 to newer standards. But we have WGET2, and Fedora 40 recently switched to using WGET2

instead of WGET." And he says: "Of course there are corner cases that break backward compatibility. Regards, Tim."

Okay. So if you see anyone running around screaming about a CVSS of 10.0 in WGET while looking up to see whether the sky is falling, you can put their mind at ease. All anyone ever had was a concern raised by seeing that semicolons were being mishandled. No exploit, no worms, no remote code anything. The CVE for this minor parsing flaw appears to have just been assigned and published this last Saturday, June 15th, so it's quite recent. NIST's National Vulnerability Database lists the CVE, but doesn't yet have any CVSS assigned. I just looked this morning.

As I was going over all this again just before the podcast, I did find a new reference at Red Hat which lists this with a CVSS of 5.4, which is far more sane. So anyway, I just wanted to put everyone's mind at rest. WGET, I'm still using I'm sure 1.something or other. But again, not in any way that's insecure, and just to - typically just to grab podcast audio once a week.

Okay. As a result of a lawsuit recently brought by Canal+, a French sports broadcaster, a French court has ordered three very popular, well known, public DNS providers - Google, we've heard of them; Cloudflare, we know them; and Cisco, oh, yes - to selectively edit their DNS domain name resolutions in order to block the lookup of around 117 individual pirate sports streaming domain names. And you know, why not just sue copper wire for having the audacity to carry electrons at this point? We've covered this sort of conduct before, and it's just as wrong now as it was then.

TorrentFreak posted an article last Thursday which explained how this battle has been slowly escalating for the past year. They wrote: "In 2023, Canal+ went to court in France to tackle pirate sports streaming sites including Footybite.co, Streamcheck.link, SportBay.sx, TVFutbol.info, and Catchystream.com. Canal+ said that since subscribers of local ISPs were accessing the pirate sites using their Internet services..."

Leo: How dare they.

Steve: I know, and how dare those wires carry those electrons with those sporting people.

Leo: Shocking.

Steve: It's just, you know, we should - oh, yeah. The ISPs, they said, should prevent them from doing so. When the court agreed, all major French ISPs were required to implement technical measures to comply. Since the ISPs have their own DNS resolvers for use by their own customers, these were configured to provide non-authentic responses to deny access to the sites in question, all 117 of them. Naturally, in response to this blackout, savvy Internet users that had not already done so simply changed their settings to use different DNS providers.

Leo: Yes, of course.

Steve: I mean, you can just imagine people texting each other, shoot, you know, Sportybites.com just went dark.

Leo: Nuts.

Steve: Oh, just change your DNS.

Leo: Fruitybits, yes.

Steve: Put it to 1.1.1, yeah, and off you go. So they just changed them to different providers - Cloudflare, Google, and Cisco - whose resolvers had not yet been tampered with, at least at that time. Use of third-party DNS providers to circumvent blocking is common. So last year Canal+ took legal action against those three popular public DNS providers - Cloudflare at 1.1.1.1, Google at 8.8.8.8, and Cisco at 208.69.38.205. hike - in each case demanding measures similar to those which had already been implemented by French ISPs. And once again the court agreed.

TorrentFreak writes that: "Tampering with public DNS is a step too far for many Internet advocates. But for major rightsholders, if the law can be shaped to allow it, that's what will happen. In this case, Article L333-10 of the French Sports Code, which became active in" - I know.

Leo: Well, there you go.

Steve: The French Sports Code. We knew that they are good sports. Anyway, which became active in January of 2022, seems capable of accommodating almost anything. TorrentFreak says: "It reads, when there are 'serious and repeated violations' by an 'online public communication service' whose main objective is the unauthorized broadcasting of sports competitions, rightsholders can demand 'all proportionate measures likely to prevent or put an end to this infringement, against any person likely to contribute to remedying it.'" So that's about as broad as any language could be.

As a consequence: "Two decisions were handed down by the Paris judicial court last month, one concerning Premier League matches and the other the Champions League. The orders instruct Google, Cloudflare, and Cisco to implement measures similar to those in place at local ISPs. To protect the rights of Canal+, the companies must prevent French Internet users from using their services to access around 117 pirate domains. According to the French publication which broke the news, Google attorney Sebastien Proust crunched figures published by government's anti-piracy agency Arcom." So using figures from their own government, Google crunched some numbers and concluded that the effect on piracy rates, if any, is likely to be minimal.

"Starting with a pool of all users who use alternative DNS for whatever reason, users of pirate sites - especially sites broadcasting the matches in question - were isolated from the rest. Users of both VPNs and third-party DNS were further excluded from the group since DNS blocking is ineffective against VPNs. Proust found that the number of users likely to be affected by DNS blocking at Google, Cloudflare, and Cisco amounts to a whopping 0.084% of the total population of French Internet users."

Then, citing a recent survey, which found that only 2% of those who face blocks simply give up, shrug, and don't bother to find other means, he reached an interesting conclusion: "2% of 0.084% is 0.00168% of Internet users. In absolute terms, that would represent a group of around 800 people across all of France," who would find that, oh, shoot, my pirate sports are no longer available, and I'm just giving up because it's just not worth typing anything into my keyboard to get it back.

They said: "In common with other courts which have also been presented with the same arguments, the Paris court said the number of people using alternative DNS to access the sites, and the simplicity of switching DNS, are irrelevant." We don't care. "Canal+ owns the rights to the broadcasts. And if it wishes to request a blocking injunction, a blocking injunction is what it shall receive. The DNS providers' assertion that their services are not covered by the legislation was also waved aside by the court. Google says it intends to comply with the order. As part of the original matter brought in 2023, it was already required to de-index the domains from search results under the same law.

"At least in theory, this means that those who circumvented the original blocks by switching to these alternative DNS services" - oh, my - "will be confronted by blocks all over again. But given that circumventing this set of blocks will be as straightforward as circumventing the originals, that raises the question of what measures Canal+ will demand next, and from whom."

So like I said, let's sue copper for having the audacity to indiscriminately carry anyone's electrons. Just as we have in the European Union, regarding whether or not and exactly how and when communications can be encrypted, here again we have another instance of a collision between the power of the courts and the power of technology. Technology desperately wants to be and to remain content agnostic. The electrons just don't care. But those who are in the position to impose their will upon the electrons only want them to carry the content they approve of. Google has capitulated, and I presume that Cloudflare and Cisco will follow suit.

Before long, DNS is going to become an even greater mess than it already is. And the most annoying part of this is that it's going to be a mess that doesn't actually solve any real problem since pirates will just switch over to some lesser known, well-off-the-map DNS provider that isn't on anyone's radar. And we should remember that DNS is really only a convenience in the first place. It's a pretty good bet that these pirate content hosting services are using a fixed IP. So just placing an entry into a machine's local HOSTS file will permanently solve the DNS problem by preventing the system from even querying external DNS. And we should also not forget that these piracy streaming sites are being hosted somewhere by someone. THEY are the true culprits, and it's they who should be shut down, not honest and well-functioning free Internet services offering DNS resolution. Wow.

Leo: It's, yeah, go after the - but this is how they always do it. And it's frustrating.

Steve: Right, because they can get to Google. Google has people.

Leo: Right.

Steve: You know? Snootystream.qr doesn't.

Leo: Piratebooty.net, no.

Steve: That's right. Well, you know, we sent them a letter, and nobody replied. The service didn't go dark. So what can we do?

Leo: Yeah, yeah, yeah. It's kind of a...

Steve: Wow. Okay. During the run-up to today's podcast, I almost - this was yesterday, or, I mean, Sunday evening - finished with the code I wanted to have in place for automating the handling of undeliverable email. So it's still the case that nothing - I just wanted to let everyone know who's been registering at GRC and subscribing to the Security Now! list, nothing has ever been sent. So the reason you haven't received anything is that I haven't sent anything. But that should start happening this week.

Among other minor announcements, I am now pre-checking the domain name portion of user entries by performing my own DNS lookup, before I try to submit an email address that might bounce. Anyway, so the point is that typos in the domain name portion are being caught before the user leaves the page. And GRC now supports plus signs in email addresses. So anyone who wishes to filter incoming email by, for example, adding a +GRC or a +SN to the end of their mailbox name may now do so. There's a simple "Delete" button on GRC's mail.htm page. So if you're enjoying using, you know, if in general you enjoy using plus signs, you can easily delete your original non-plused email account and create a new one with a plus.

Leo: Yeah.

Steve: So for anyone who wants. And I have three quick pieces of useful feedback. Tallis Blalack, he wrote: "Long-time Security Now! listener and SpinRite owner. What was the program you used to download your email into something you could easily search? You mentioned it on an SN episode a few years ago. My domain host originally offered free email. The cost went to $4.80 per year when my email storage went over 2GB, and I was willing to pay that instead of making the time to reduce my email size. Now they've moved to an email-as-a-service and have increased the cost to $48 per year. It's time to back it up and clean it up as I move to a new hosting service. Thanks for all you do."

So Tallis, this is my periodic opportunity to share one of my best discoveries ever. It's the free and wonderful MailStore Home, which is MailStore's free personal Home edition. As I said, it remains one of my most valuable discoveries which I'm happy to recommend without reservation. Through all the years I've been using it, it has never let me down. After installing it, you can aim it at your existing email archive or service, and it will suck it down, index it, and build an instantly keyword searchable, you know, substring searchable local database.

The way I have my own personal email system setup is that all incoming mail to me is also cloned into a separate "archive" email account, and everything I send is also automatically blind copied into the same archive account. That way the archive receives a copy of everything coming in and going out. Then in the very early morning, at 3:30 a.m., a Windows Task Scheduler task fires off, starts up MailStore to retrieve, index, and permanently add to its archive everything in my day's worth of email. So it's always kept up to date within a day.

And astonishingly, all of this is 100% free. And I should note that, while I've never looked into it, they also have enterprise solutions available for a refreshing one-time purchase, no subscription. For $260 you get an enterprise-wide email archiving and retrieval solution that integrates with Microsoft 365, Google Workspace, and other email providers. You know, and even whatever you are using yourself. So again, MailStore is the company, just at MailStore.com. They're a German outfit. And the Home version is completely free. And, you know, just hats off to them. It's a beautiful solution.

Eric, a listener of ours who has some firsthand info about how IT goes over at the New York Times. Remember last week we talked about them losing, was that 270GB of, like,

basically they lost everything, all of their code, all of their website technology, everything. So he says: "Hello, Steve. I want to share a comment regarding the New York Times, and a bit of history. Around 2010 I was working for a company that provided endpoint security, and New York Times was our customer. They were stuck on an old, unsupported version of our software. Despite all the advances in behavioral-based machine learning and non-signature-based detection technologies, they insisted on running as 'AV only.' We had countless emails and phone calls documenting our strong recommendation that they upgrade and apply the full security stack which they were entitled to."

Leo: Aha. This is 14 years ago.

Steve: Yup.

Leo: Holy cow.

Steve: He said: "The recommendations were ignored, and they were successfully attacked. They proceeded to publicly name us as their security provider and the technology that failed. Of course we could not go public with a big 'We told you so,' and we were forced to take the black eye to protect the customer relationship. So with whatever cybersecurity incident happened to the New York Times recently," he said, "I do not believe a word they say. I have no doubt the story about the company's IT guy leaving the private GitHub access token exposed..."

Leo: Yeah, blaming somebody, yeah.

Steve: "...is only a cover story for a far worse problem." So anyway, Eric, thank you for sharing that. This really makes me wonder how many of the problems we examine each week are effectively self-imposed.

Leo: Right.

Steve: You know, we hear about a critical shortage of qualified IT professionals. But I suppose there's no shortage of unqualified IT wannabes. It would be interesting to know what the real reason was for them not wanting to improve their own security when an improvement was being offered and even pushed. From what Eric described, it sounds like it wasn't money because they were already purchasing technology they refused to deploy. Incredible.

Mark Newton said: "Steve, I was searching through the show notes. I was thinking you mentioned having a reMarkable 2 or something similar, and how much you liked it. It appears there are a couple of different manufacturers out there. I thought you specifically mentioned the reMarkable 2, or perhaps their version? You would not believe how often the word 'remarkable' applies in the notes from multiple Security Now! podcasts." I guess I'm saying remarkable, remarkable, remarkable. There's three. Anyway, he said: "You sparked my interest. Pricey, but it looks like it would be helpful."

Okay. So Mark, Leo turned me onto the reMarkable 2 stylus-based eInk tablet, and I never want to use anything else. Okay, now, in fairness, I did once feel the same way about the Palm Pilot.

Leo: How many do you still have in the fridge?

Steve: I don't know. Extras of them went into the fridge for long-term storage and availability and safekeeping, and they've never come out. So instead I have iPhones and iPads that didn't exist at the time. So I can't say something better may not come along someday. So I suppose the lesson is to never say never.

But what I can attest to is that, through the years since college, I kept trying over and over to use every spiffy new technology to replace my longstanding use of those wonderful light green quadrule engineering pads with the grid printed on the back side of the paper. Those pads with a soft lead mechanical pencil were always the ultimate solution when I was brainstorming or working on a problem. When I'm coding, I'll often make sure I don't make one of those common "off by one" errors by drawing simple diagrams of reduced complexity cases and then check it as I'm running the code in my head.

After 50 years with those green engineering pads, the reMarkable 2 tablet finally replaced them for all time. I mean, it is absolutely my go-to tablet. I love it. So, yes, the reMarkable 2 is the device.

And Leo, after sharing our last sponsor relationship with our listeners, we're going to discuss, dare I say, "The Angle of the Dangle."

Leo: You dare not.

Steve: Also known as the saga of BingoBango.com.

Leo: Okay, well, we'll find out. We'll learn more.

Steve: The dangerous dangling domains.

Leo: The dangerous dangle of [crosstalk]. We're doing different software, doing a different studio, different cameras. And I apologize because I don't have a mic on/off switch. So everybody said, "Hey, Leo, you cracked a beer during that last segment." It wasn't a beer, I promise. This is zero-calorie sparkling water, okay? Tropical fruit flavor, if you must know. And by the way, before the next show, yes, we're going to have a mic switch. Every once in a while Leo needs to slurp.

All right, Steve. When I heard the name of this show, I said, I don't think Steve knows what that means. But then I forgot, you're a boomer. You know what it means. You know who doesn't know what it means, all of those people watching us now, and there are almost 500 of them on YouTube, the same number on Twitch, on our Discord channel for the Club. We've got a big audience today. Thank you, all of you, for joining us. I appreciate it. And maybe it has something to do with the title. I don't know. What do you think? Steve?

Steve: If so, we may be disappointing some of our viewers.

Leo: I think so.

Steve: Or maybe not because, you know, sometimes you don't want any dangle.

Leo: Yes.

Steve: Actually, as a general rule, especially when DNS is concerned, dangling is not good.

Leo: No.

Steve: As I said, this week's Picture of the Week presents a step-by-step flow of how Domain Takeover happens. Even though it's entertaining, even though I understood how this happens by the time a listener sent me that fun diagram, untangling the diagram was more work than following an explanation of how this happens. So I want to thank everyone who wrote with their explanations.

Since I love my rack of router, firewall, switch, and server hardware at Level 3, as I noted at the start last week, I've never so much as glanced at the architecture of cloud-based hosting. Just has never been on my radar. Maybe in another 15 years, when I can no longer lift a fully loaded server into the rack, but not yet.

So the moment a listener mentioned that the trouble was almost certainly with a DNS CNAME record, the entire story became clear. And since we've somehow never talked in detail about how DNS CNAME records are used, and how badly they can go wrong, I thought this would be the perfect opportunity.

All of this, of course, is in reference to last week's topic, the rise and fall of code.microsoft.com, and specifically to how it happened that a malicious actor managed to, however briefly, install their own presence into the "code" subdomain of microsoft.com. The root of the problem is the potential for any organization's DNS records to point to something, anything, that they do not themselves control because, if they do not control the resource that their DNS records point to, the presumption would be that someone else may. Which brings us to the question, "How could that possibly happen?" Which many of our listeners will recognize as being the close cousin of "What could possibly go wrong?"

The first thing to appreciate is that this general problem has existed since the beginning of the Internet, when DNS was first created. So the issue of anyone's DNS pointing to anything they don't control itself is not new. And that's never been good. But what is new is the way cloud-hosted resources have evolved; and how, as a result of this, the old problem of DNS misconfiguration has quickly become much greater.

Okay. So let's take the fictitious site BingoBango.com. The people running it decide that they're fed up with running their own website and other services. It's time to move to the cloud. And for the sake of this discussion, they've chosen Microsoft's Azure web hosting service.

When Microsoft created Azure, they obtained the domain azurewebsites.com to act as the root anchor for all of their subscribers' services. So when someone wishes to have Azure host a site and their services, they register with Azure and choose a name for their service. It might be a human readable and meaningful name, like BingoBango, or it might be just anything that's available like BB123. Regardless of what they choose, their new site is reachable as a subdomain of the azurewebsites.com domain. So it could be reached at Bingobango.azurewebsites.com or, in the case of not caring much about the site's subdomain name, BB123.azurewebsites.com.

But the BingoBango folks already own BingoBango.com, and no one wants to type "bingobango.azurewebsites.com" every time they want to visit the BingoBango website. Microsoft was well aware of this, as was Amazon and all the other cloud providers. They knew that the names of their services, and its subscribers' subdomains did not matter because DNS could be used to hide whatever they were.

We're all aware of the way good old standard DNS IP address lookup works; right? A DNS typical IPv4 "A" address record or IPv6 "AAAA" address record is used to turn a DNS name into one or more IP addresses. When a user's client wishes to connect, it uses DNS to look up the domain's IP address and then connects to one of the IP addresses returned. So it's a one-step process. But the success of this one-step process requires that the server's IP address does not change.

If Microsoft could guarantee a fixed IP address for their subscriber's website bingobango.azurewebsites.com, then standard DNS A-record lookup could be used. Just like the BingoBango people once pointed their BingoBango.com IP address at their own server, they could instead point their BingoBango.com IP at the IP address provided by Microsoft. Then visitors to BingoBango.com would simply go to the server at the IP provided by Microsoft.

But DNS provides a better and much more flexible way to achieve the same goal which allows cloud providers total freedom in the way they set up at their end. For example, it might be useful to allow different regions of the world to be accessing cloud resources at different IP addresses so that offering a single fixed IP to everyone would be less convenient. So this is made possible by a different type of DNS record known as a CNAME. The "C" of CNAME stands for canonical.

Whereas a DNS "A" address record returns one or more IP addresses, a CNAME record returns another domain name. In programming we would refer to this as introducing a level of indirection because, instead of a domain name pointing directly to an address record, it points to another domain name which then probably points to an address record. And this domain name indirection using DNS CNAME records is the way cloud hosting providers are typically accessed.

Using our example, once the BingoBango people have established their services at bingobango.azurewebsites.com, they replace their DNS's BingoBango.com "A" record with a CNAME record pointing to bingobango.azurewebsites.com. Now, anytime someone wishes to visit the BingoBango.com website, their web browser will look up BingoBango.com and will find a CNAME record there. That CNAME record tells the browser that the name they were looking up, BingoBango.com, is actually a "canonical name" for the actual hostname, and that CNAME record provides the actual host name, which is bingobango.azurewebsites.com. So THAT is the domain name that it should pursue in its quest to find an IP address for its connection. The user's DNS resolver then looks up the IP for the domain bingobango.azurewebsites.com, obtains that IP from the azurewebsites.com DNS servers, and returns that to the user for their connection to the BingoBango.com website.

What's significant here is that two DNS resolutions were required, and that the second lookup was for a property located under the azurewebsites.com domain. This allows Microsoft to host their subdomains at whatever IP addresses they choose and affords them the flexibility to change this at any time they wish. The BingoBango.com DNS simply refers anyone who asks to the bingobango.azurewebsites.com DNS for its IP.

This CNAME resource creates a very powerful system of DNS pointers; and, as we know, with great power comes great responsibility. So it should come as no surprise that things have not always worked out well. The problem occurs, as happened to Microsoft with their code.microsoft.com subdomain, when an organization deletes the hosted services and domain being pointed to by some other canonical DNS name. At that point the canonical name is said to be left "dangling" because it's pointing to a nonexistent host name. The bad news is that this opens the way for someone else, who would almost always have to be malicious, to re-register their own cloud service under the same domain name as what was previously deleted.

In the case of the code.microsoft.com subdomain, the subdomain would have had a DNS CNAME record. When that record, code.microsoft.com, was retrieved, it would have provided the name of the host that had been deleted. So the bad guys would simply register their own host under that CNAME record's name, and anyone then accessing code.microsoft.com would be referred to their probably malicious services. And note that their servers would appear, and this is important, the malicious servers would appear to any web browsers to be within microsoft.com. This creates additional potential for various browser domain origin trickery.

Three and a half years ago, on November 25th of 2020, "The Daily Swig" cybersecurity news and views newsletter at PortSwigger.net posted a piece under the title "Rampant CNAME misconfiguration leaves thousands" - they could have said hundreds of thousands - "of organizations open to subdomain takeover attacks."

They wrote, and this is three and a half years ago: "Security researchers have discovered more than 400,000 subdomains with misconfigured CNAME records, leaving many at risk of malicious takeover as a result. When websites are externally hosted, the CNAME (Canonical Name) record used to map their canonical domain and subdomains to the third-party host domain, this means that the canonical, rather than the host domain, appears in the browser's address bar.

"Pinaki Mondal, of the security firm RedHunt Labs based in India, wrote in a blog post that when a cloud-hosted website is deleted, but the DNS entry pointing to the resource is retained, attackers can potentially re-register the host, add the organization's subdomain as an alias, and thus control what content is hosted under the original canonical name. Attackers can then serve malicious content to visitors and potentially intercept internal emails, mount PDF-based click-jacking attacks, hijack users' cookies and sessions by abusing OAuth whitelisting, and abuse cross-origin resource sharing to harvest sensitive information from authenticated users." In other words, you really don't want bad guys to appear to be in your own primary domain.

"Using a tool," they write, "that conducts mass DNS resolution, RedHunt Labs found more than 424,000 subdomains with misconfigured CNAME records during an automated trawl of 220 million hosts. The number of sites that were abandoned, for example, if they belonged to defunct organizations, was unclear due to the additional need to look up company registries to obtain that information. But by adding HTTP response grabbing, the researchers uncovered evidence that 139 of Alexa's top 1,000 domains may have fallen prey to subdomain takeovers." 139 out of the top 1,000.

"RedHunt Labs identified 33 third-party services that allowed for potential subdomain takeovers. With nearly 63% of vulnerable DNS records pointing to Shopify, most

vulnerable domains belonged to ecommerce operators. Landing page creator Unbounce accounted for the second highest number of vulnerable domains at 14%, followed by Heroku at 10%, GitHub Pages at 4%, and BigCartel at 2%. Drilling into the data, RedHunt said 'www' was" - wow, that actually makes sense; right - "was the most frequently vulnerable subdomain." So think about that.

Leo: Wait a minute. Wait a minute. You can have a www - oh, .bingobango.com. Okay.

Steve: Exactly.

Leo: Yeah.

Steve: So BingoBango.com might still have their email being handled by their own BingoBango.com, but www...

Leo: .BingoBango...

Steve: You know I used that name just because it was so fun to say it.

Leo: Well, you saw that our chatroom found somewhere you can register it for a mere $8,000. So go for it.

Steve: Oh, that's interesting. Because when I looked last night it was at $15,000.

Leo: Yes, that's what Galia said. So maybe he got a deal, he found a deal.

Steve: Nice.

Leo: Half off, Steve.

Steve: Now, I have to tell you that my first thought was to go with BingoTips.com because I thought, okay, how can you have Bingo tips? But there is a BingoTips.com.

Leo: Of course there is. Dude, you're quoting security research from a company called PortSwigger.

Steve: Yeah, okay.

Leo: Whose number one [off mic]. I'm sorry, I pushed the wrong button. Whose number one product is Burp Suite. Although I've got to say there's a certain consistency to their naming.

Steve: Right, they're staying faithful to the name.

Leo: They're actually legit, but I just love it.

Steve: Oh, I know they are. That's why I quoted them is they're a real group.

Leo: Yeah. It's hysterical. Oops, I've got to zoom you. Here we go. Nope, not that. No, no. Whoops, not you. No, no. Here. Wait a minute. There you go.

Steve: Command Nancy, Command Nancy.

Leo: Command Nancy. Okay.

Steve: Okay. So it totally makes sense that www would be the most frequently vulnerable subdomain because somebody who doesn't want to host their own website would send www off to some third party. So, yeah. Wow.

This guy Pinaki Mondal at RedHunt pointed out that by suppressing and removing the "www" - remember when Google did this? - and the "m." subdomains from the Chrome browser's address bar, that was at Chrome 69 onwards, he said: "Google had inadvertently made it more difficult for users to determine whether they 'might be browsing attacker-controlled content.'

"RedHunt found around 200 non-functional .gov site subdomains with misconfigured CNAME records, and one had a 'wildcard' CNAME record, which poses a particularly dangerous security risk. And Mondal noted that prestigious universities owned some of the roughly 1,000 misconfigured .edu subdomains." And anyway, I guess my point here is Microsoft certainly is not alone in having tripped over this problem.

"The findings show that despite the potentially calamitous impact of subdomain takeovers, many well-resourced large organizations are struggling to comprehensively discover and track their own ever-expanding infrastructure." You can just imagine what the DNS must look like of some of these organizations. Yikes. "Mondal also noted that Roblox, Starbucks, and the U.S. Department of Defense are among the organizations to have remediated subdomain takeover flaws through HackerOne in the past year." That was two and a half or three and a half years ago.

The article ended with The Daily Swig noting that the year before they posted this, in 2019, it had previously reported on subdomain takeover flaws stemming from Windows 8 Live Tiles feature, and the year before that a misconfigured Microsoft subdomain, back in 2018. So not even Microsoft's first subdomain problem. And four years before this article, which is from November of 2020, and before RedHunt's research, researchers from the University of Delaware and the College of William and Mary published their research titled - and this is back in 2016 - "All Your DNS Records Point to Us - Understanding the Security Threats of Dangling DNS Records." So my point being, not a new problem. Very powerful, but very easy to get wrong.

So our takeaway for today's podcast is that Microsoft definitely made a serious mistake when they left their "code" subdomain dangling, but they are far from the only organization to have ever done so. In today's increasingly complex IT landscape of

overlapping services, where such services are increasingly being outsourced, DNS can become quite complex and convoluted. And we once talked about how one of the tricks being used to track website visitors is to point a CNAME record of a website to an advertiser's domain as a means of giving them first-party cookie-tracking status. That struck us as being ultra slimy at the time, but it's the times we're living through.

So just a heads-up reminder to any of our listeners who have responsibility for the management of DNS within their organization. You likely already know that DNS is not as simple and straightforward as it once was. It might be worth making sure that you know who any of your organization's CNAME records are pointing to, and that they have a good, ongoing, and legitimate reason for being granted such a potentially powerful position since they're sharing your organization's root domain.

Leo: Wow. Absolutely a chilling thought. Although if you've figured out how to get BIND working, you probably know enough to keep yourself from getting CNAME poisoned.

Steve: Well, I don't know. You know what happens is it's one little thing after another. You add this. You add that. It gets increasingly complex. And then, like, someone doesn't renew a contract; right?

Leo: Yeah, right.

Steve: And so they disappear, but it doesn't automatically delete that record from DNS. DNS is still pointing to them, even though they're no longer offering the service.

Leo: It's amazing. What a world we live in. Steve, you're the best. Thank you for putting up with this crazy situation here. We're trying out new software, trading in the TriCaster for a software-based solution called Restream. Which unfortunately allows me to put chat comments into your feed.

Steve: I just decided I'm not going to try to read them while it's going on.

Leo: Don't look at them.

Steve: I can't do that.

Leo: And I'm having a little fun with it, but so are the chatters. It also gives us the opportunity to stream this video into multiple places once again, which is great. We've got Twitch watching. There's 445 people on YouTube, 138 people on Twitch. We can expand it to beyond that, to X and to Insta. You know, I think this is a really nice way to be able to share what is such an important show with a broader audience. So thank you.