The Hertzbleed Attack

Description: This week, after dealing with a major piece of errata from last week, we look at Germany's reaction to the EU's proposed "Let's monitor everyone and privacy be damned" legislation. The Conti gang finally pulls the last plug. We have an update on the status of Log4j and Log4Shell and a weird proposal for a "311" cyberattack reporting number. A sweeping 56 new vulnerabilities were found and reported across the proprietary technologies of major industrial control technology providers. And this week we have a piece of miscellany, followed by 10 interesting items of closing-the-loop feedback to share from our listeners. We will then take a deep dive into the latest "Hertzbleed Attack" which leverages the dynamic speed scaling present in today's modern processors. We'll examine another effective side-channel attack, which is even effective against carefully written post-quantum crypto and can be used to reveal its secret keys.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-877.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-877-lq.mp3

SHOW TEASE: Coming up on Security Now! it's me, Jason Howell, filling in for Leo just for this week, joining Steve Gibson, of course, who does the majority of the talking because he is the security pro. We say farewell to the Conti gang, at least I think it's a farewell. Also we look at the 311 cyberattack reporting number. Don't worry, it has nothing to do with a rock band. Also we check in on the still-going Log4j and Log4Shell vulnerability. Steve reminds us of, you remember Heartbleed? Yeah, we take a look back in time, eight years ago, to Heartbleed. And then dives deep into the latest Hertzbleed attack. You're not going to want to miss that. Steve Gibson explains it all next on Security Now!.

JASON HOWELL: This is Security Now! with Steve Gibson, Episode 877, recorded Tuesday, June 28th, 2022: The Hertzbleed attack.

It's time for Security Now!. This is the TWiT show where we talk about all the latest security news, the security happenings, happening all around the Internet right now and beyond. Of course I am not Leo Laporte. I am Jason Howell, filling in for Leo this week. And I'm so excited anytime I get the chance to sit in with Steve Gibson. Welcome to the show, Steve.

Steve Gibson: Hey, Jason. Great to have you back. You are our fill-in when Leo is on a cruise.

JASON: Or not on a cruise, but somewhere else.

Steve: Or off traipsing around the East Coast.

JASON: Yes, exactly.

Steve: Wherever.

JASON: Yeah, wherever he happens to be. I'm happy to be here, thank you.

Steve: It's always good and comfortable to have you. So I'm glad for that. So this is our last episode of the middle of the year, 877 for June 28th. This is a topic, our main topic has been on my radar for a couple weeks. In fact, I sort of forgot about it. It fell through a crack last week and what we had to sort of talk about. This one is titled "The Hertzbleed Attack." And our longtime listeners, I think it was from 2008 - no, I know, that's too far away. But maybe '13. Anyway, I've got it in the show notes. We'll talk about it when we get there. There was the famous Heartbleed attack. And this is clearly a play on that. Which, and this is really interesting. I think that our listeners are going to get a kick out of it.

But we've got to first deal with a major piece of errata from last week. Then we're going to look at Germany's reaction to the EU's proposed - basically it amounts to "let's monitor everyone and privacy be damned legislation." The Germans are not too happy about that. The Conti gang has finally pulled its last plug. We've got an update on the status of the Log4j and Log4Shell exploits. Also a weird proposal that seems to be actually gaining some traction for a "dial 311" cyberattack, like, incidence report number. It's like, okay. You know, I dial 911 for problems. 611 I think is for your phone to be repaired. And now we've got 311. So it's okay.

There's also a sweeping 56 new vulnerabilities that were found and reported across the proprietary technologies of a handful of major industrial control technology providers, and a little bit of philosophy of what's wrong with the way we're doing things there. We have a piece of Miscellany. And then there was just a lot of great feedback from our listeners since last week. So we have 10 interesting items of closing-the-loop feedback from our listeners.

And then we're going to take a gratifyingly deep dive into this latest and interesting new Hertzbleed attack, which manages to leverage the dynamic speed scaling present in today's modern processors. It's yet another way of leaking information, believe it or not. It's a side-channel attack. And it's even effective against post-quantum crypto. So nowhere to hide there. And after you tell us about our first sponsor, I have to say we've got a really fun Picture of the Week. We'll save it till then.

JASON: We'll expand when it's time to expand.

Steve: So I think another great podcast for our listeners.

JASON: Absolutely. And it took me back, hearing Hertzbleed and thinking of Heartbleed, because I totally remember when that was a big deal. That was, I'm just looking on Wikipedia right now, released 10 years ago, discovered eight years ago. So that sounds about right.

Steve: Okay.

JASON: Yeah, I remember that being a pretty darn big deal, so I'm super curious to hear how that ties into now. All right. Picture of the Week time. This is how we start the show. Tell us what I'm looking at here. This is actually - this is a big picture. It like scales three pages.

Steve: Well, that's the point, exactly. This is the first time in 17 years that the Picture of the Week did not fit on one page. A clever blogger created this, sort of a joke, but also to prove a point. We've often talked about how OpenSSL has become a, well, a mixed blessing. It is literally the Swiss army knife of secure communications. It's also the Swiss army knife of secure communications even if that's not what you want. If you just want to connect two endpoints using TLS, well, it's not clear that OpenSSL is the best thing to use. And it's a coincidence that it was a bug in OpenSSL that caused the Heartbleed attacks and that whole vulnerability.

So, okay. What people are looking at, if you have the show notes, is if OpenSSL's command line options had a GUI. That is, if you - because, I mean, the command line options are just nuts for OpenSSL. So the person who did this blog posting, basically to prove a point, he gave OpenSSL a GUI. And so, for example, it starts of course with just, let's see, one, two, three, four, five, six lines of tabs. So you have a tabbed user interface.

And for example, there's CA for Certificate Authority. Ciphers. I see CRL, that'll be Certificate Revocation List. I see dhparam, that'll be Diffie-Hellman parameters for signing. DSA, that's Digital Signature Algorithm. Dsaparams. There's EC params, so Elliptic Curve parameters. There's encode, engine. What else do I see? Generate RSA keys, OCSP, that's Online Certificate Status Protocol. Password, who knows what that is. Generic pkcs7. Public key, private key, pkeyutil, something about prime numbers, a random number generator, rehash, there's RSA, I mean, then there's s_client and s_server to create a secure client or secure server connection. Okay. Those are just the tabs.

JASON: Right.

Steve: Now, the last tab in all of that, just alphabetically sorted, is x509, which is the standard that's been defined for the format of certificates. So then the panel of this GUI begins with input format, is it a PEM or a DER. Output format, PEM or DER. And a button to open the input file, a button to open the output file. And I'm not going to enumerate all these. But anyway, the point is here we have then buttons and sliders and forms to fill out, and checkboxes, do you want this, like text output options, print serial number value, print subject hash value, print issuer hash value, print subject DN, issuer DN, email address, certificate purpose, blah blah blah blah blah. I mean, endlessly. And these are all reflected in the command line of this insane OpenSSL that has just been growing and growing and growing since it was first created.

So anyway, I knew that we've often talked about OpenSSL. And I loved it that after three pages of scrolling through this, you get down to the bottom, where the guy who created this added a note. He said: "Note that even this is incomplete. It covers about 80% of one corner of OpenSSL's functionality. The certificate policy options have a lot more knobs that were not included." It's like, oh, my god. Anyway, just a perfect Picture of the Week for our podcast, and I wanted to share it with our listeners who would get a kick out of it.

JASON: I love it. And I love that what you're seeing, because the first thing that my eyes go to, right, is just the sheer length of it. And then when you go back up to the top and you realize, that's just one tab of, I don't know, if I had to guess, like 30. It's pretty remarkable.

Steve: Well, and it is an accurate reflection of this great, I mean, you can do anything with OpenSSL. It'll create certificates. It'll verify certificates and validate them. You can create, I mean, it is literally it's like the one place for doing everything. And the problem is that it is - oh, the other thing is that it's where this stuff is first tested. So if someone's

creating like a new, well, for example, TLS. OpenSSL went through a series of versions, went to 3.0, and then SSL 3.0 became TLS 1.0, and then 1.1 and 1.2 and 1.3. It supports all of that, and it's where these things were first tested. And after they were tested, they may get spun off into something else, but they also stay there.

So anyway, it is a remarkable utility, but I would not recommend that anyone actually use it these days. There are a number of very lightweight, purpose-specific TLS libraries. If all you want to do, and that is what most people simply want to do, is to create a solid robust TLS connection between two points, the last thing you want to do is load OpenSSL into your server process in order to make that happen.

Okay. I have to start with a piece of errata from last week. I don't know how to explain or apologize or what, the brain fart that I had. Last week I talked about Firefox's Total Cookie Protection. And then with Leo we went to GRC's Cookie Forensics page, and there I expressed my puzzlement over the fact that it didn't seem to be working. Well, it was working perfectly. That is, everything was working perfectly. Firefox's Total Cookie Protection was turned on and working perfectly and just the way it should be. And my page was working correctly, even though I thought something was broken. So I just got tangled up, and I want to fix that.

The brilliance of Mozilla's approach is that third-party cookies do still work just as they always did. So the fact that I was getting red, like, warnings of oh my god, this site GRC has active third-party cookies from GRCtech.com, is yeah, of course. That's third-party cookies. Total Cookie Protection doesn't stop third-party cookies, it creates the context. It binds the third-party cookie, a specific third-party cookie, to the domain where it was sent. So this is the way it's going to look.

Basically the short version is my Cookie Forensics page was never designed to test for third-party cookie sequestration or context. In order to do that, I would need a third domain. I don't know what I would call it, but I don't have one. I mean, I have one, but I'm not going to take the time to go do that right now. I'm working on SpinRite, and nothing will distract me from that. But I just wanted to clarify that everything's working. Mozilla's got it going. Firefox is protecting people. And the point is that third-party cookies have not been suppressed.

It is the case now, although my Cookie Forensics page does not illustrate this, if GRC exchanged third-party cookies with, as it does GRCtech.com, were you to go to GRD.com and try to get GRCtech.com's cookie, your browser, which does have a GRCtech.com cookie from GRC.com, would go, huh?, if you were at GRD.com, and it would say, no, you don't have a cookie for that. And if GRD.com set a third-party cookie for GRCtech.com, then the browser at GRD.com would say, yeah, okay, fine. We've got a third-party cookie now for GRCtech. But it would be completely disconnected from and distinct from the third-party cookie of the same name, which you transacted over at GRC.com. And that's Total Cookie Protection. It means that third-party cookies work as they always did, but they cannot be used to track. They are, I mean, just elegantly and beautifully blocked from tracking.

And as I said last week, the only thing that I don't understand, except I obviously didn't even understand how my own Cookie Forensics test worked, I don't understand how it took us this long to get such an elegant and simple solution. It must be that there is just tremendous lobbying pressure, like there's got to be tremendous pressure on the browsers. Well, we know that Google is about tracking, so they're not going to fix this in Chrome any sooner than they have to. But here's Mozilla off to the side, wanting to be the alternative browser that we all love because it and Safari and now everything else is Chromium. So anyway, it's working great. I just wanted to correct the record. I just got myself tangled up last week.

Okay. Also last week, we talked on Tuesday that there was going to be a LastPass webinar two days after that on Thursday, where LastPass's gurus, the CTO, the chief tech guy, and what we found was a bunch of marketing people, of course. That's how you pay for getting the technical information is putting up with that. We're all going to hold a webinar. And they did. The hope was that they would clarify the very confusing blog post they'd made the previous week where they talked about passwordless logon coming soon to LastPass, and then said, oh, and kind of FIDO2, but blah blah blah. Like, what?

Anyway, so I sat through this mostly marketing spiel, and here's what we learned. I snapped two slides from it. The first slide was titled "Also, it's a Journey..." And under that it says "Real talk: truly removing passwords from the login equation could take years. Why?" And then it says: "Universal passwordless access requires FIDO2 support from devices, operating systems, browsers, and authenticators, but also from each website individuals need to access."

So, yeah, that's what we've been talking about. The last couple weeks, ever since Apple's announcement of passwordless and passkeys and their support for FIDO2 coming in iOS 16, it's like, okay. And Google has said they're going to do it, as has Microsoft. And we've identified two issues. One is it's very clear that each of these three providers is kind of rubbing their hands together saying, oh, goodie, we're going to sync the passkeys you create in our little ecosystem among the devices that share that ecosystem, but we're not talking yet about whether we're going to let them go, other than one at a time if you move it to a different device.

And in order for this whole passkeys thing to work, every device that a user uses has to be dynamically, that is, in near real-time, synchronized with every other so that if you create a new passkey with a site on one of your devices, you can later go to a different device and log on as you, and that passkey will be known with the client that you're logging in from. And that requires, because they didn't use SQRL, which doesn't have any of this problem, but fine, that requires that in real time you cross-synchronize all the devices.

There's been, last week we saw, we learned first-hand from the president of FIDO and the chief something or other, both of them said, oh, no, we don't know how to do that. FIDO2 doesn't do that. We haven't thought about that. So use it now, and we'll worry about that if it becomes a problem. Well, we already know it's going to be a problem. So that's the first thing.

And the second thing is, yes, you have to have web server support. I mean, when we get iOS 16 and whatever - you're the Android expert, Jason, whenever Google is doing that in Android. Well, we'll be - I loved my analogy last week. I said, it's like the person who invented the first shortwave radio. Hello? Can anyone hear me? Wait, there's only one radio. So no.

JASON: It's going to take a little bit more than just that. Nice thought, but...

Steve: Yes, exactly. So anyway. So yes. So basically LastPass is saying, okay, you know, like cool your jets. We know everybody wants FIDO2. But when you get it, there's nobody to talk to. So we're going to move in that direction. We're heading there. But we don't have it yet. And in fact that brings us to the second slide which showed their three phases. Phase 1 is Available Now! and it has an exclamation point because it's like, Now! And that is "Passwordless login to the vault using LastPass Authenticator." Which, okay, that's good. So you put your thumb on the thumbprint reader, or you smile at the camera on your smartphone. And ooh, look, I'm in. So, okay. That's good. Doesn't seem that hard. Everybody has that. Oh, yeah, and it's available now in LastPass.

Then later this year, no exclamation point on that one because not yet. They say: "Adding FIDO2-supported Authentication [which is to say] security keys and biometrics, for additional security and flexibility when using passwordless login to your vault." And I'll remind everyone from my digging around in Bitwarden, as I mentioned last week, Bitwarden already offers the use of FIDO2 login. Bitwarden, by the way, a sponsor of the podcast and TWiT Network. They've already got it. So LastPass will be later this year, catching up with Bitwarden, basically the idea being more protection using the FIDO2 passkey technology to more strongly protect access to your own vault of old-school usernames and passwords. This has nothing to do with logging onto websites using FIDO2. It's the other way around. You're logging onto your vault using FIDO2 with some sort of Authenticator.

And finally Phase 3, which is what nobody has yet because, again, we've got the chicken-and-egg problem. With Phase 3 they said: "Coming in 2023." In other words, next year sometime. And they said, and this is from their slide: "Adding secure storage of passkeys, in addition to passwords" - meaning LastPass will do both, it'll do old-school usernames and passwords, and you can also put passkeys in there, which it'll unlock and treat the same way and allow you to then use your passkeys with FIDO2-compliant websites when and to the degree that they start to emerge.

And of course that's what Apple, Google, and Microsoft have all just announced their support for. Neither LastPass nor Bitwarden, nor as far as I know any other password manager offers a solution for that today. And why would they? As I said, one short-wave radio. Can't talk to anybody. So we don't know when in 2023. Probably they'll set the priority based on what traction these passkeys and FIDO2 is appearing to get over time. And so I think that timing probably works well.

The point, the reason we're spending so much time focusing on passkey support for FIDO2 is that it isn't - "it" meaning LastPass or Bitwarden or 1Password or whatever - isn't Apple, Google, or Microsoft, where their intention seems to be to use the fact that there is some inherent lock-in to their own competitive advantage. The password managers will inherently be cross-platform. And so I've been cautioning our users, our listeners of the podcast for a couple weeks now, careful. Before you invest heavily in one of these isolated platforms, see about exportation of passkeys because you may end up wishing that you were on some platform that was dynamically synchronizing.

And again, the dynamics is the key. Even if Apple were to - if Apple, Google, and Microsoft or even the FIDO were to get together and come up with a passkey blob export/import, that doesn't sound dynamic. That is, for this thing to be practical, all of these ecosystems would need to be dynamically sharing passkeys so that if you create one on your iPhone, when you sit down at your Linux desktop, the Authenticator there has received the new passkey from Apple somehow. That's just never going to happen. It's like, they're not going to talk to each other on the fly. That's just - no. So it's going to be a third-party password manager that solves this problem.

Okay. Politico's headline on this was Germany forces EU into damage control over encryption fears. And then they added that Berlin has criticized the EU's plan to fight child sexual abuse material, CSAM, as a threat to privacy and fundamental rights. And, you know, no kidding. We talked about this earlier this year when the EU announced what they were going to do. I guess, well, yeah, earlier this year. I guess it was about a month and a half ago. That's when the European Commission proposed a new law to crack down on sexual abuse of children online.

And as our listeners will recall, that proposal was controversial, to say the least. It is clearly facing some serious headwind from the bloc's largest member country, Germany. Since the proposal first aired, the Germany government has repeatedly slammed the proposed legislation as an attack on privacy and fundamental rights until finally, just last

week, Germany's Digital Minister Volker Wissing warned that the draft law, as he put it, "crosses a line."

Berlin's opposition prompted the EU's Home Affairs Commissioner to step in last week in what seemed like an effort to limit the damage with some sort of appeasement. Although, if you look at exactly what the guy said, it's not clear that worked. This was some guy named Johansson. He defended the proposal during an impromptu press conference, so I guess you could forgive him for maybe not being prepared, which occurred in Luxembourg on last Friday afternoon, saying that the legislation "is much more targeted" than the current regime to scan for illegal images, which is not what the legislation said. He said: "Will only allow only to do detection after a court decision or another independent authority have decided so after consultation with data protection authorities and with specific technologies that have been approved." What?

Okay. Those were direct quotes. Again, off the cuff. I'm not sure what he was meaning or meant to say. But if we were to take it as gospel, whoever wrote that is apparently describing some very different legislation than the text we carefully examined more than a month ago. The intent of that legislation was very clear, as was the breach of end-to-end encryption that would be needed to make it work.

And this is devolving into the sort of mess that it was bound to. Sweden's EU commissioner spoke alongside Germany's Interior Minister Nancy Faeser, insisting upon Faeser's "strong support" as a mother, an adult and a politician. Okay. Not clear what she's strongly supporting. For her part, Faeser said that "the initiative, from the German point of view," she said, "we support this," but added that "for us, it's important to find the balance," meaning between the right to confidential communication and cracking down on child sexual abuse material. The problem is it's just not possible to provide both at the same time. It's not.

So to remind everyone, as it was presented, the revised rulebook wants to force all tech companies, including messaging app providers WhatsApp, Apple's iMessage, Instagram and Telegram, and I'm sure Signal is there, too, anybody, to scan, remove, and report illegal photos and videos of this CSAM material. Courts could also order digital companies to hunt down manipulative conversations between potential sex offenders and children, which is referred to as "grooming."

So no one wants our technology to be used for criminal child sexual abuse. But there's no way around the fact that examining photo and video content, and scanning and interpreting text messaging proactively while looking for suggestive "grooming language patterns," there's no way around the fact that it can only be accomplished by defeating both the spirit and the act of end-to-end communications privacy. You can't have it both ways.

So while welcoming stronger action to protect victims of online abuse, a big collection of German government ministers has nevertheless piled on to lament that the EU proposal would effectively result in mass surveillance of people's private messages and thus undermine encryption. Of course. Again, you can't do it both ways. So we know that crypto technology is obedient. It'll do anything we want it to. But what we want, apparently, is to invade everyone's privacy on the off-chance of detecting that their conduct might be criminal, while at the same time invading no one's privacy. Well, you can't.

JASON: It doesn't sound likely.

Steve: It's just no. And as we've said more broadly, not just the EU particular legislation, more broadly, this is the big dilemma. Governments want the ability to see what's going on. And users - and the government says they don't want to violate end-to-end

encryption. Oh, no, no, no, you can still encrypt things. But also not. What? Okay. So the great debate, the decryption debate of this century is how do we settle this.

And Jason, you may still be alive by the time that happens. I'm feeling great, I'm healthy and happy, but I don't know how we're going to resolve this.

JASON: I know. Artificial intelligence.

Steve: Ah. That's right.

JASON: That's the blanket that we throw on every problem nowadays and hope.

Steve: Send all that to Skynet. And what's weird is that, of all the companies, I would argue that Apple's suggestion was the least privacy intrusive. What Apple wanted to do was to take, although it wouldn't be completely effective, they wanted to take hashes of known objectionable material and put all those hashes on people's phones and check any image being sent against the hash, meaning hash the image, see if it's in the list, and only if so, then flag it for close scrutiny by a human to decide if it's a false positive or not.

And everyone, oh, my god, no. They objected. And I get it. They objected to the idea of even hashes of images being on their own handset. Yet it was a clever way of solving this problem of addressing the need to in some way deal with the problem of CSAM while not actively monitoring everyone's content. Basically, and I'm reminded of this because of your comment about AI, it's sort of a local AI-ish sort of solution. But no. I mean, that thing was just dead on arrival.

JASON: Well, and not only that, I mean, it was also an understanding problem, to a certain degree; right? Like there are a lot of people that - like in our circles, hash makes sense.

Steve: We know what a hash is.

JASON: We know what a hash is. The majority of people, when the read the article about what's changing on their iPhone, and they see the word "hash," like they're not going to understand the actual reason that it's a hash or the difference between storing a hash of images on your phone and storing actual image data of some way, shape, or form. You know what I mean? So I hate to call it a marketing problem, but ultimately, probably for the wide majority of people, that's exactly what it was.

Steve: Yeah, I think you're right. It was a marketing problem. I think that's the right term.

JASON: Yeah.

Steve: Okay. So the Conti gang have finally pulled the last plug. Recall that we previously noted the clear indications that the very prolific, actually too prolific for its own good, Conti ransomware gang had boxed itself into a corner earlier this year after Russia invaded Ukraine by clearly and forcibly siding with Russia in Russia's very unpopular, unprovoked war. The sanctions that the West and most of the rest of the free world leveled against Russia choked off Conti's victims' ability to pay ransom to Conti in Russia even if they wanted to.

So in reaction, Conti set up a shell game. All the members of Conti abandoned ship except for one last guy who remained behind to keep the fires burning. This member continued leaking data and taunting Costa Rica, which was nominally their last victim,

which created a faade of Conti still being a going concern and a running operation, while all of the rest of Conti's members quietly moved on to other ransomware groups.

Last month's report by Advanced Intel stated that the only goal Conti had for this final attack, that is, the Costa Rica attacks, was to use the platform as a tool for publicity, to keep the spotlight on Conti as they performed and faked their own death, keeping their multiple rebirths in other names and under other names off anyone's radar. Even though they were pretending to still be active as Conti, the ransomware operation was not performing any further attacks, and the data being leaked by this one remaining Conti member turned out to be from earlier attacks. To confuse researchers and law enforcement even more, this sole Conti member released the same victim's data on both their site and Hive's data leak site, where he is also an affiliate. But this was just all a charade with the rest of the Conti ransomware crew infiltrating or even taking over other ransomware operations.

Well, as of last week, finally, the masquerade is over, and the Conti ransomware operation has finally shut down its last public-facing infrastructure, which consisted of two Tor servers which were used to leak data and to negotiate with victims. And of course no victims remain. So they're gone. The Conti name has been retired, and its crew has spread out and around and have taken up operations under other names. These guys know what they're doing. So they formed a potent crew. And it would be good if they just ended up being less powerful in the future.

JASON: All right, Steve. What do we have up next? Oh, the Log4j, Log4Shell. This seems to just like never go away, as well. It's kind of like the Conti gang.

Steve: And in fact that's exactly the point.

JASON: Yes.

Steve: Last Thursday CISA posted a useful reminder report that I thought was interesting on several levels. Its title was "Malicious Cyber Actors Continue to Exploit Log4Shell in VMware Horizon Systems." I didn't have it in the show notes, but VMware Horizon basically is a remote desktop system, the idea being that in the cloud you spin up instances of Windows, and you run Windows remotely in a remote desktop mode, or also apps. So it is inherently public facing in order to function. And VMware knew right off the bat they had a problem. They had it patched immediately. Of course, as we know, it's one thing to patch it. It's a different thing for those patches to be deployed.

One of the reasons I thought this was important is because it's easy for us to become complacent. For a few weeks late last year, Log4j and Log4Shell were big news and everyone was scurrying around fearing that this might result in the meltdown of the Internet, or another big worm, or the end of civilization as we know it. Okay, probably no one really believed that. But it was initially grabbing all of the tech press headlines, and it certainly had our attention on this podcast.

Then we learned an important lesson. Because it did not enable super-simple drop-and-go exploitation, despite the fact that the presence of the Log4j vulnerability remained quite widespread remember we talked about how long it would take for all of the Java packages that were dependent upon it to be updated, if ever, I mean, we're talking years the mass of the world's script kiddies did not pick it up and run with it. Potentially widespread though it was, actually exploiting the Log4j vulnerability turned out to require significant per-instance work. It wasn't the lowest hanging fruit, so lower hanging fruit continued to be preferentially exploited.

But as we also said several months ago, since the Internet is still here, neither did the Log4j threat go away completely. It was quietly adopted and added to the toolkits of the

world's most sophisticated threat actors for judicious deployment when and where its presence had been overlooked on a system where it might offer a way in. And that's what CISA's posting last Thursday was intended to remind us of, and to document.

CISA wrote: "The Cybersecurity and Infrastructure Security Agency (CISA) and United States Coast Guard Cyber Command (CGCYBER) are releasing this joint Cybersecurity Advisory (CSA) to warn network defenders that cyber threat actors, including state-sponsored advanced persistent threat actors" - meaning the big guys - "have continued to exploit CVE-2021-44228 (Log4Shell) in VMware Horizon and Unified Access Gateway servers to obtain initial access into organizations that did not apply available patches or workarounds."

Okay, now here we are at the end of June. Log4Shell was December of 2021. So we're now six-plus months downstream, and attacks, effective attacks are still happening, not just against random stuff that nobody knew there was a problem with. No, against VMware's Horizon systems, which are known to be vulnerable, which VMware immediately created patches for And here we are six-plus months later, and those systems have not been updated. Thus they're still vulnerable, and bad guys are getting in and setting up shop.

They said: "Since December 2021, multiple threat actor groups have exploited Log4Shell on unpatched, public-facing VMware Horizon and Unified Access Gateway servers. As part of this exploitation, suspected advanced persistent threat actors implanted loader malware on compromised systems with embedded executables enabling remote command and control. In one confirmed compromise, these APT actors were able to move laterally inside the network, gain access to a disaster recovery network, and collect and exfiltrate sensitive data.

"This Cybersecurity Advisory provides the suspected APT actors' tactics, techniques, and procedures" - a new term, TTPs - "information on the loader malware, and indicators of compromise (IOCs). The information is derived from two related incident response engagements and malware analysis of samples discovered on the victims' networks."

So I have a link in the show notes to the entire Advisory, which it does have a lot of really interesting details, I mean, specific executables, DDLs, scripts, the IP addresses of these command-and-control servers, where all of this stuff is linked to and communicating with. For anyone who wants more detail, as I said, a link in the show notes. But the moral of the story is an important reminder, which is that exploits which invariably fall off the radar don't cease to exist. Just because old problems are no longer being actively discussed should not be any source of comfort for anyone. So definitely worth keeping systems up to date. And, boy, the lesson is you just have to wonder who these machines belong to. They were anonymized, that is, the CISA advisory does not tell us who got themselves hacked by waiting at least six months to update their VMware Horizon system. So certainly not any of our listeners. We know that.

Okay. Last Wednesday, in their third meeting, a group of cybersecurity company experts met in Austin, Texas - that's last Wednesday - to provide industry recommendations to CISA. So they have sort of an industry meeting group, government-facing interface. The group was founded in June of last year, and its first meeting was held in December. This one last week was its third. The group is split into six subcommittees, each focused upon different issues which cover cyber workforce; information dissemination; cyber hygiene, whatever that is; technical advisories; critical infrastructure; and misinformation.

The cyber hygiene subcommittee, led by Apple's VP of Corporate Information Security, suggested that CISA "launch a '311' national campaign, to provide an emergency call line and clinics for assistance following cyber incidents for small and medium businesses."

The measure was also floated by the communications subcommittee, which was led by a member of Tenable's board.

A spokesperson for Checkpoint, the security firm Checkpoint Software, said that the idea for a "311" emergency line is "smart and timely." The Checkpoint executive noted, he said: "Right now we're seeing on average organizations in the United States being attacked 868 times per week." 868 known attacks per week. "The emergency line," he said, "can make for a faster path towards incident response." During this month of June, which is almost over, Checkpoint said that they were seeing an average of at least 27 cyberattacks against small and medium-size businesses per week, and that this was an increase of 72% compared to last year. So the attacks are on the rise, and there's lots of them. And when you think about it, there are far more small and medium-size businesses than big behemoth businesses. And while the small/medium-size businesses can't fork up so much ransom as a biggie, there's lots more of them.

Okay. So meanwhile, CISA executives and others continue to push for more robust and reliable incident reporting. As we covered at the time, cyber incident reporting legislation was passed and signed into law earlier this year, but it only covers critical infrastructure organizations. All qualifying organizations are now required by law to report breaches to CISA within three days, 72 hours, and report ransomware payments within one day, 24 hours.

Three weeks ago, during the annual RSA security conference, Eric Goldstein, CISA's executive assistant director for cybersecurity, spoke at length about how damaging the lack of data on ransomware attacks in the U.S. is for organizations like his. And I assume that more attacks might mean more funding, although CISA has recently been getting a great deal of funding. Eric told attendees of RSA that, he said: "Only a small fraction of ransomware victims are reported to the government, and the problem is getting worse." He said: "We have no idea what the actual number is. We have no idea what level of ransomware instructions are occurring across the country on any given day."

And I hadn't really stopped to think about it, but that would really be true; right? I mean, if you aren't required by law to report a breach of your organization, and if you don't have cyberattack insurance, as I imagine most small-to medium-size businesses probably don't, so you're able to keep the fact of an attack closely held, then why would small businesses do anything more than arrange payment and hope to obtain a recovery key? I can see how that lack of information flow to CISA would be frustrating for them.

But even so, it's unclear what good calling some 311 cyberattack line would do. It's not as if the government is going to pay the ransom for you. And you can imagine that the bad guys are going to take the standard action of anyone doing extortion of saying "Don't get law enforcement involved, or you'll never see your data again. We'll know if you do." So I suppose it's unsurprising that small- to medium-size organizations that are attacked are not reaching out, and that CISA is annoyed by this because they'd just like to know.

Some reports are just demoralizing and depressing. One such is a recently published report from Forescout which details a collection of 56 vulnerabilities which they have found in the so-called "OT" or what's known as Operational Technology category of devices. OT is the newer term which we once used for SCADA systems. So they're the technologies used to monitor and control oil and gas refining, chemical and nuclear power generation and distribution, manufacturing, water treatment and distribution, mining, and building automation. You know, the industrial control sorts of things.

What's interesting is that these products are sold under the rubric "secure by design," and many even carry, I think three quarters even carry certifications for OT operational technology, like having met operational technology security standards. But Forescout's

report was titled "OT:ICEFALL," which is the name they gave to this collection, they said "A Decade of Insecure-by-Design Practices in OT."

The vulnerabilities affect Siemens, Motorola, Honeywell, Yokogawa, ProConOS, Emerson, Phoenix Contract, Bentley Nevada, Omron, and JTEKT. And the disclosure of this mess was coordinated with CISA and other relevant government agencies around the world, other governments' equivalents of CISA. Summarizing what they found, the vulnerabilities could be divided into four categories: Insecure engineering protocols, so like they made up their own protocol, and it was not secure. Shock. Because we know how difficult it is to create secure protocols. Or weak cryptography or broken authentication schemes. Again, if you roll your own good, good luck. Third, insecure firmware updates. And fourth, remote code execution through native functionality. In other words, remote code execution was supported natively by some of these things.

38% of the 56 vulnerabilities allowed for compromise of credentials. One in five, 21%, allowed for firmware manipulation; and 14% allowed remote code execution. And despite that, three quarters, as I noted, of the affected product families carry some form of feel good, yet apparently worthless, security certification. Forescout explained their intentions and why they felt this work was important.

They wrote: "With OT:ICEFALL, we wanted to disclose and provide a quantitative overview of Operational Technology insecure-by-design vulnerabilities rather than rely on the periodic bursts of CVEs for a single product or a small set of public, real-world incidents that are often brushed off as a particular vendor or asset owner being at fault. These issues range from persistent insecure-by-design practices in security-certified products to subpar attempts to move away from them. The goal is to illustrate how the opaque and proprietary nature" - and you can almost hear my voice in this - "the opaque and proprietary nature of these systems, the suboptimal vulnerability management surrounding them, and the often-false sense of security offered by certifications significantly complicate OT risk management efforts."

So this report highlights another of the recurring themes of this podcast. And that is that something is very wrong with the model we currently have for proprietary technology being blindly applied in critical infrastructure such as power generation and water treatment without any truly effective oversight over the security of these proprietary systems. We need to move away from this model and away from a culture where the security status of products whose security affects vast numbers of people is allowed to be obscured by their proprietary intellectual content. This is the classic voting machine problem. Until this happens, until the culture and I guess it's going to have to be government who says, you know, if we're allowing you to manage our water, then we're not going to allow you to use systems that have not been opened and scrutinized. And again, a voting machine. You can sell the hardware, but you can't keep the software proprietary. It's just too important how it works. And until that happens, our infrastructure's going to be fragile, essentially by design.

JASON: I don't understand why that's not the default.

Steve: I know. Isn't it dumb?

JASON: Doesn't make any sense, yeah.

Steve: And Jason, when you think about it, though, it is the culture. Every software license that we click on says that the manufacturer is not responsible no matter what it does. The Windows license, well, you know, we don't know if it even works. We don't know if it's good for anything. And you're going to click this to eliminate all responsibility from us. You're responsible for what it does. I mean, that's just like saying, oh, well, gee, I want that, like to use it, so what choice do I have? Okay, fine.

JASON: Yeah, exactly. And all it is is a single click away; right?

Steve: Yeah. And has anyone ever successfully read that crap? You can't. I mean, your mind goes numb.

JASON: Yeah, for sure.

Steve: Trying to read and parse all that boilerplate. Okay. I have something completely off the reservation for Miscellany, totally random and off-topic. I tweeted about it. I just wanted to share it with our listeners. And it would be completely off-topic if it didn't have a little bit of time-travel in it, even though it's not sci-fi. It is something that my wife and I discovered completely by chance Sunday evening. It's a low-budget, independent, 90-minute, well-written romantic comedy. And believe me, I'm not a customer normally of romantic comedies. It just doesn't get me. But this one on Netflix did. It's called "Long Story Short." It was released a year ago on July 2nd of 2021.

It isn't any sort of blockbuster. And interestingly, it only scored a 6.5 on IMDB, which is below my normal 7.0 threshold for IMDB. But it's odd because all the reviews that I saw were like raving about it, with 8's, 9's, and 10's. So in any event, this little gem carries and I think beautifully delivers an important message about the way we conduct our lives, and not in a preachy way. So anyway, it's called "Long Story Short." Since I know how widely people's tastes vary, I'm not offering a guarantee. But I know that I would have been glad to have been told about it. So I didn't want to pass up the opportunity to share this little discovery. Again, "Long Story Short" on Netflix. And if you have someone to watch it with, so much the better.

JASON: I'll check that out. Yeah, you had mentioned that in email, and I'm trying to know enough about it to pique my interest, but not so much about it that I'm going to spoil the movie because it seems like a movie you probably don't want to spoil very much.

Steve: You don't want to spoil it. I'll just say that I am a fan of dialogue.

JASON: Yeah, me, too.

Steve: I loved back in the day "The West Wing" because of the Aaron Sorkin, like real people don't actually talk this way kind of dialogue. The lead male actor, you would almost think he's a standup comic because his lines are delivered with such timing. And, I mean, you have to - don't watch this if you're like falling asleep because it really requires your attention. But it's very clever. And again, the point of it is, not only was it fun, but it's a worthy message is bound into this thing. So I think you and wifey would like it as much as I and wifey liked it.

JASON: Right on. Thank you, Steve. I'll check it out.

Steve: Okay. So 10, a surprising number of closing-the-loop bits of feedback. Lawn_dart, he tweeted: "Just listened to this week's SN, and the encryption problem still isn't quite complete. If an attacker knows a significant amount about the plaintext, then encrypting plaintext guesses might be more efficient." So he's saying going forward encrypting plaintext guesses. He said: "The puzzle only truly works if the plaintext is also indistinguishable from entropy."

Now, I'm sharing this tweet because what you always do with public key encryption - although this was symmetric encryption, so not public key. So I guess what he's saying is that if you were guessing what the plaintext was, then you could use it to guess the keys. The problem is, well, I guess the problem is the plaintext is likely far longer than

the private key that you're trying to guess. So there are, like, there are already 2^256 times 2^256 guesses in double encrypting with two symmetric keys. So 512 bits. But plaintext is going to be way longer than that, typically. It's going to be a book or a movie or something, megabits in length. So I don't think that's a practical solution.

Tiemo Kieft, he said: "Routers are IOT devices. They interact with the physical world through LEDs. Either that, or smart light bulbs are not IOT." And of course he's referring to what we talked about, the NIST formal definition of IoT as being something that had a sensor or was able to affect the environment in some fashion. And, you know, yeah. And routers have light bulbs on them, or LEDs, as well. So okay.

Arvind, wow, his last name is Narayanan. His Twitter handle is easier, it's @random_walker. He said: "I mis-clicked on one of my 150 open tabs" - meaning he has a browser like I do, he says - "and it happened to be a tab that's been open since 2019 with a paper that has a solution to the exact research problem I've been puzzling over today." He says: "This is the moment I've been waiting for, and I've decided to never close a tab again." So anyway, just a bit of humor on the topic of, yes, keeping a gazillion tabs open and arranging to make that practical.

LDizzy tweeted: "Hello, Steve. I've been listening to a lot of the talks lately about passkeys and how it locks you into the ecosystem you start with more or less. But I think there's a big chunk we're missing here. While the public/private key pair does function much like a more sophisticated password, where things will get better in sites is when they accept more than one passkey. Instead of having a single password that's reset via email when lost, sites will most likely allow for the addition of several passkeys. If I am an iOS mobile user that also runs Windows on my desktops, which I am" - much as I am - he says, "then I could add an iOS-based passkey on my phone that's synced via iCloud. Then, whenever I get around to logging into the site from my computer, I could use my iPhone once to authenticate and add a second passkey from the desktop that would then be synced within Microsoft's ecosystem."

He says: "This makes sense to me and seems like a new paradigm. There's no real reason why sites do not allow multiple passkeys, although this wouldn't have been logical with the old username/password combo." Okay, and I don't understand why that's true, but okay. He says: "The downside is that if you use multiple ecosystems, then you'd have to add a passkey within each. But they don't have to be synced between ecosystems, and it's a one-time thing for each. I doubt many people use more than two. I hope I'm not being overly optimistic, and I'd appreciate your take on this. Love your work."

So I guess my main take is everything about what LDizzy tweeted presumes that sites will allow multiple passkeys per account. And I see no reason for that to be true. Granted, that would be cool. But then, if you do that, I guess if you were to change a passkey, it would change the passkey that you used when you logged on, so the site would have to know which of multiple passkeys you used when you logged on. And then how would you control access to the other passkeys? It sounds like a little bit of a logistics problem there. I'm not sure, but we'll find out once sites begin to support them. I don't know, I've not looked yet in depth into WebAuthn. And I am thinking that we're going to have to do a podcast, a deep dive into the operation of WebAuthn since it will before long, I think it's inevitable, become the way websites authenticate. And so we certainly need to cover that on this podcast.

Joseph Fienberg tweeted: "My wife and I each received an email about the Facebook tracking class action settlement. When I went to click on the link to the file containing the class action claim, UBlock Origin blocked lzzgcc5d.r.us-east-1.awstrack.me, which is not secure and is a tracking link for a tracking settlement where we'll each probably get 50

cents. I thought Security Now! listeners would get a laugh out of that and the irony of this." Indeed. Thank you, Joseph.

Thomas Martin said: "Hi, Steve. I wanted to weigh in on your solution to the double encryption dilemma in SN-876. Your position was that an attacker cannot know the intermediate text, and therefore the two encryptions cannot be attacked separately. The conclusion was then that double encryption was substantially more secure than single encryption. If you can brute force a single 256-bit cipher using AES-256 in one day, then it does not take two days to brute force double AES-256, it would take on the order of 2^256 days." Agreed.

He says: "Unfortunately, this line of reasoning is incorrect" - oh - "as it does not take into consideration another type of attack, a meet-in-the-middle attack. The attacker needs one known ciphertext-plaintext pair. Meaning one instance where the ciphertext and the plaintext were encrypted. They encrypt the plaintext with every possible key and store the key and the text pairs in a lookup table." And then he notes that it requires 2^256 operations and on the order of 2^256 storage. "They also decrypt the ciphertext with every possible key. Each resulting text is checked against the lookup table. If it matches a stored text, then the attacker has a pair of keys that decrypt the known ciphertext to the known plaintext. This would need to be confirmed with other plaintext/ciphertext pairs, and there is a possibility of a coincidental match."

Okay. Thomas is correct as far as it goes. And so the point was that a meet-in-the-middle attack is absolutely worth discussing. It is a means of producing a computational and storage-based shortcut to the brute forcing approach which is what we've talked about. That is, where you would need to try the first decryption of which there's 2^256 possible keys, then try all of the 2^256 possible second encryption keys for each of the first, thus making the total effort 2^512.

There is a better way, and Thomas highlights it. It's a famous meet-in-the-middle attack where if you can get a ciphertext and plaintext pair, and if you have sufficient storage, and this is where this falls into impracticality, unfortunately, meet-in-the-middle attacks were famous back when key lengths were 2^64 or thereabouts. Then you could kind of say, okay, well, it might be feasible to have that much storage. This requires 2^256 sets of keys and matching text which can then be compared after the first guess. But each of those guesses needs to be compared against all of the 2^256 sets of storage. And at that point I think that this sort of brute forcing shortcut, while theoretically possible and worth reminding ourselves it exists, now that keys have gotten as long as they have, you just - yes. If it were possible, it's a shortcut. So Thomas, thank you for it.

Douglas Nichols says: "Double encrypting can be divide-and-conquer if you are using an authenticating encryption like AES-GCM or CCM modes on at least the outer encryption since the integrity check will fail." And I thought that was a great observation. All of the assumption that we've been making is that authenticating encryption is not there, specifically because it isn't, like, hard to do. If you immediately get confirmation when you've guessed the proper first half of the key, then in fact it would take you two days, if it took you one day to solve the outer. So Douglas, thanks for pointing it out. It's not the problem we're solving, but it is also definitely worth noting.

Sean OBrien said: "Many years ago you gave almost the opposite answer to the divide-and-conquer attack. Back then you said the encryptions add because of authentication. If the first encryption is signed, then the first decryption is obvious. It's in the case without authentication that the first decryption results in noise." And so Sean, thank you for reminding me that I did mention basically exactly what Douglas had tweeted, and that is that if you have any sort of authentication of the first encryption, then it's trivial to determine when you have guessed right with the first one.

Rando said: "A better proof that we live in a simulation is Murphy's Law. Because statistically it is unlikely for everything possible to go wrong all the time without exception," he says, "but in my experience that's exactly what happens."

JASON: Awww.

Steve: So I got a kick out of that.

JASON: I'm sorry, Rando.

Steve: The fact that you have Murphy's Law means that, okay, yeah, you're right. That's a failure of stochastic randomness. Can't happen in a completely random environment. So we must be living in a simulation with a bad random number generator.

And the guy who we mentioned last week who once was testing Windows in their hardware-based Windows software testing lab, sent me a series of tweets. The first one said: "Hey, Steve. It was really cool hearing you talk about my Microsoft career as a Senior SDET." And that was a Device Engineer of Testing. That's the DET. He says: "I love sharing my experiences with the Windows Hardware Testing Lab and ESC teams responsible for testing the WINMAIN branch and talking about why the current system post-2015 layoffs is so bad." So it was nice to have the loop closed with that.

JASON: All right. So why does it Hertz so bad, or so good? The Hertzbleed Attack, what have we got going on here?

Steve: So this has everything you want. It's got a snappy name.

JASON: Yeah.

Steve: It's got its own website. It's got a memorable logo, you know, everything the modern vulnerability needs. And its name is an obvious play on 2014's Heartbleed attack, for good reason. Remember that eight years ago Heartbleed was one of those rare vulnerabilities, like Dan Kaminsky's realization about DNS spoofing vulnerabilities, that truly moved the industry to action. That one was CVE-2014-0160. Ah, those good old days of four-digit CVEs. Yeah, we don't have that anymore.

Heartbleed, that is, eight years ago the original Heartbleed attack, its summary reminds us, says: "The Heartbleed Bug is a serious vulnerability in the popular OpenSSL" - as I mentioned before about OpenSSL - "cryptographic software library. This weakness allows stealing the information protected, under normal conditions, by the SSL/TLS encryption used to secure the Internet. SSL/TLS provides communication security and privacy over the Internet for applications such as web, email, instant messaging, and some virtual private networks.

"The Heartbleed bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. This compromises the secret keys used to identify the service providers and to encrypt the traffic, the names and passwords of the users, and the actual content. This allows attackers to eavesdrop on communications, steal data directly from the services and users, and to impersonate services and users."

And finally they said: "We have tested some of our own services from attackers' perspective. We attacked ourselves from outside without leaving a trace. Without using any privileged information or credentials, we were able to steal from ourselves the secret keys used for X.509 certificates, usernames and passwords, instant messages, emails, business-critical documents, and communications."

Okay, so that was eight years ago with Heartbleed. Today, of course, we have Hertzbleed. And anyone who is interested, it's Hertzbleed.com - H-E-R-T-Z-B-L-E-E-D dot com. Okay. So here's how Hertzbleed, today's problem, describes itself: "Hertzbleed is a new family of side-channel attacks: frequency side channels. In the worst case, these attacks can allow an attacker to extract cryptographic keys from remote servers that were previously believed to be secure." And I'll note that whereas Heartbleed leveraged a bug in the OpenSSL library at a certain version level until it was patched, this is not a bug. Not a bug.

"Hertzbleed," they said, "takes advantage of our experiments showing that, under certain circumstances, the dynamic frequency scaling of modern x86 processors depends upon the data being processed. This means that, on modern processors, the same program can run at a different CPU frequency and therefore take a different," they called it "wall time," you know, like real-world time, "when computing, for example, 2022 plus 23823 versus 2022 plus 24436."

And they finish: "Hertzbleed is a real and practical threat to the security of cryptographic software. We have demonstrated how a clever attacker can use a novel chosen-ciphertext attack against SIKE" - S-I-K-E, which I'll describe in a minute - "to perform full key extraction via remote timing, despite SIKE being implemented as 'constant time.'"

Okay. So SIKE stands for Supersingular Isogeny Key Encapsulation. And all we really need to know is that it's a state-of-the-art post-quantum crypto. In other words, meant to solve the quantum computer problem. This is like next-generation. It's been applied for NIST certification. But what it is is not really that important. The point is that, regardless of how secure something is, and this SIKE, S-I-K-E, is like post-quantum super secure, its keys can still be extracted. And if that's the case, it's game over. Doesn't matter how secure and how far post-quantum you are. If you've got the keys, doesn't matter.

Okay. So I should first say a little bit about power consumption in modern state-of-the-art semiconductor systems. Today's CPUs still use transistors. But whereas the first transistors that were originally invented were current amplifiers, which are known as "bipolar transistors," today's transistors use electrostatic charge rather than active current to open and close their switches. They are metal oxide semiconductor field effect transistors. So that's the abbreviation M-O-S-F-E-T, or MOSFET.

If a MOSFET transistor is either on or off, no power is consumed. None. In order to "flip the switch," the controlling gate of a field effect transistor must either be charged up with electrons or drained of its electron charge. But once that's done, the gate will remain charged up or drained out, and the switch will remain open or closed without consuming any power.

So the key thing to appreciate is that in a large array of interconnected MOSFET transistor switches, which is any modern CPU, power is only consumed when the states of those MOSFET switches are changed. And the amount of power consumed is proportional to how many switches are changed and how often they're changed.

This explains why we can hear our laptop fans spin up when our machines get busy, and why they eventually spin back down some time after our computers have been idle. When a CPU is idling, it is doing less work because it's got less work to do. Since switching transistors on and off is what consumes power, batteries can be made to last longer and systems can be made to be more "green" by slowing down the clock speed of CPUs when they don't have much work to do. A reduced clock speed means fewer transistors switching per second, which means less power needed and consumed, which means less power lost as heat, and fans don't need to spin as fast since there's less heat needing to be removed from the CPU.

The dynamic speed scaling of today's CPUs has become a virtual art form, with the CPU tightly and instantly changing its speed based upon the instantaneous demands that are placed upon it. And if you've guessed that this dynamic speed changing results in a new form of side-channel information leakage, give yourself an "A" and move to the head of the class.

Looking again at what the researchers said in their summary, they said: "Hertzbleed takes advantage of our experiments showing that, under certain circumstances, the dynamic frequency scaling of modern x86 processors depends upon the data being processed." In other words, and they went on to talk about how adding two different numbers would actually result in the CPU running at a slightly different speed.

So this work was done through a collaboration of a team of six researchers from universities in Illinois, Texas, and Washington. Their paper, which is titled "Hertzbleed: Turning Power Side-Channel Attacks Into Remote Timing Attacks on x86," will appear in the 31st USENIX Security Symposium being held in Boston from August 10th through the 12th in about a month and a half, this summer.

So what does this mean for us? They did a fun Q&A, which I will share bits of and also add some commentary to. First question: "Am I affected by Hertzbleed?" Answer: "Likely, yes." They said: "Intel's security advisory states that all Intel processors are affected." They said: "We experimentally confirmed that several Intel processors are affected, including desktop and laptop models from the 8th through the 11th generation Core microarchitecture." Although Intel was even broader, saying, yup, got them all.

"AMD's security advisory states that several of their desktop, mobile, and server processors are affected." They said: "We experimentally confirmed that AMD Ryzen processors are affected, including desktop and laptop models from the Zen 2 and Zen 3 microarchitectures." They said: "Other processor vendors, for example ARM, also implement frequency scaling in their products and were made aware of Hertzbleed. However, we have not confirmed if they are, or are not, affected by Hertzbleed."

So next question: "What is the impact of Hertzbleed?" They answered: "First, Hertzbleed shows that on modern x86 CPUs, power side-channel attacks can be turned into even remote timing attacks, lifting the need for any power measurement interface. The cause is that, under certain circumstances, periodic CPU frequency adjustments depend upon the current CPU power consumption, and these adjustments directly translate into execution time variations. Second, Hertzbleed shows that, even when implemented correctly as constant time, cryptographic code can still leak via remote timing analysis. The result is that current industry guidelines for how to write constant-time code, such as Intel's, are insufficient to guarantee constant-time execution on modern processors." And that is the brilliance of what these guys have done.

We talked a long time ago about the need for constant-time execution of cryptographic algorithms. Or being a bit more specific, never having any secret information, like the cipher's keying material, directly controlling the execution path through the cipher algorithm. As we know, in modern processors, all execution paths leave breadcrumb trails in the underlying microarchitecture, things like the processor modifying its future branch predictions based upon past actual branches taken and not. So constant-time code is carefully designed now to take the same path and to execute in the same number of CPU cycles specifically so as to give attackers who may be carefully watching from the outside, or even in an adjacent VM, or sharing hardware, living in a different VM on the same chip, keeping them from seeing anything about what specific data was processed.

What these guys brilliantly realized was that there's actually another form of information leakage occurring from Intel's x86 and AMD's Ryzen processors. Even though the number of CPU cycles may be constant, and the execution path may be invariant, the exact

number of individual transistors whose on and off states were changed during the computations will almost necessarily differ depending upon the data that was actually processed. And since CPUs dynamically scale their speed based on power consumption, the actual time required to perform the computation, not in CPU cycles but in actual passage of world time, will be subtly altered. And believe it or not, that can leak secret key information, and they proved it. Their code is open source and available for people to play with.

So they ask themselves: "Should I be worried?" And they answer: "If you're an ordinary user and not a cryptography engineer, probably not. You don't need to apply a patch or change any configurations right now. If you are a cryptography engineer, read on." They said: "Also, if you're running a SIKE decapsulation server, make sure to deploy the mitigation described below."

They ask: "Is there an assigned CVE for Hertzbleed? Yes. Hertzbleed is tracked under CVE-2022-23823 and CVE-2022-24436 in the Common Vulnerabilities and Exposures (CVE) system."

"Is Hertzbleed a bug? No. The root cause of Hertzbleed is dynamic frequency scaling, a feature of all modern processors, used to reduce power consumption during low CPU loads and to ensure that the system stays below power and thermal limits during high CPU loads."

"When did you disclose Hertzbleed?" And they said: "We disclosed our findings, together with proof-of-concept code, to Intel, Cloudflare, and Microsoft in the third quarter of last year, , and to AMD in the first quarter of this year, 2022. Intel" - I love this. "Intel originally requested our findings be held under embargo until May 10, 2022." So just last month. "Later, Intel requested a significant extension of that embargo, and we coordinated with them on publicly disclosing our findings on June 14, 2022." In other words, we said no. You already had six months. What are you going to do?

And the question: "Do Intel and AMD plan to release microcode patches to mitigate Hertzbleed? No. To our knowledge, Intel and AMD do not plan to deploy any microcode patches to mitigate Hertzbleed. However, Intel provides guidance to mitigate Hertzbleed in software. Cryptographic developers may choose to follow Intel's guidance to harden their libraries and applications against Hertzbleed. For more information, we refer to the official security advisories, Intel and AMD."

"Why did Intel ask for a long embargo, considering they're not deploying patches?" Answer: "Ask Intel."

"Is there a workaround? Technically, yes. However, it has a significant system-wide performance impact. In most cases, a workload-independent workaround to mitigate Hertzbleed is to disable frequency boost. Intel calls this feature 'Turbo Boost,' and AMD calls it 'Turbo Core' or 'Precision Boost.' Disabling frequency boost," they write, "can be done either through the BIOS or at runtime via the frequency scaling driver. In our experiments, when frequency boost was disabled, the frequency stayed fixed at the base frequency during workload execution, preventing leakage via Hertzbleed. However, it is not a recommended mitigation strategy as it will significantly impact performance. Moreover, on some custom system configurations with reduced power limits, data-dependent frequency updates may occur even when frequency boost is disabled. In other words, as with Spectre and Meltdown, this is another instance where a longstanding CPU optimization must be discarded if its exploitation is to be completely eliminated. Nothing is safe."

And then: "What is SIKE? SIKE," they answer, "Supersingular Isogeny Key Encapsulation, is a decade-old, widely studied key encapsulation mechanism." In other

words it's a public key sharing system. They said: "It's currently a finalist in NIST's Post-Quantum Cryptography competition. It has multiple industrial implementations and was the subject of an in-the-wild deployment experiment. Among its claimed advantages are a well-understood side-channel posture. You can find author names, implementations, talks, studies, articles, security analyses and more about SIKE on its official website."

Then they ask: "What's a key encapsulation mechanism?" Then they say: "A key encapsulation mechanism is a protocol used to securely exchange a symmetric key using asymmetric public key crypto."

They ask: "Is my constant-time cryptographic library affected?" And they answer: "Affected? Likely yes. Vulnerable? Maybe." They answer: "Your constant-time cryptographic library might be vulnerable if is susceptible to secret-dependent power leakage, and this leakage extends to enough operations to induce secret-dependent changes in CPU frequency. Future work is needed to systematically study what cryptosystems can be exploited via the new Hertzbleed side channel."

And finally: "Do you release the source code of the Hertzbleed attack? Yes, for full reproducibility. You can find the source code of all the experiments from our paper at the link https://github.com," and then the rest of the link is /FPSG-UIUC/hertzbleed. It's all there.

So we have a new attack. Much as with Spectre and Meltdown, there is no good solution. Even in the instance, and in fact it was Intel's, Intel provided a constant time crypto library, but it was constant in cycle count, not in actual real-time. Whoops. So that's going to be a problem. What you would have to do, you know, you could say, well, you could disable CPU scaling, speed scaling, during the cryptographic operation and then reenable it. That won't work because the cryptographic operation itself is what consumes a variable amount of power. So then if after the operation is complete you reenable scaling, the CPU will still immediately scale based on the power that was previously consumed.

What you would have to do is literally develop another technology, much as we now have constant time crypto, you would literally need constant power crypto so that the power being consumed by the architecture of the CPU was not affected based on the secrets that you're trying to keep in the crypto. And I would imagine that's what the ultimate solution will be is we'll get post-quantum that is not only constant in time, but constant in power consumption. And I'm glad I don't have to write that code.

JASON: If you had to choose your favorite, Heartbleed or Hertzbleed, what would it be, Steve?

Steve: Well, Heartbleed's solved.

JASON: Right, yeah, right.

Steve: Hertzbleed is an open problem now, which makes it really more interesting.

JASON: This is the challenge of the work that you do. There's always a new thing. So even when these things happen, and it seems like the end of the world, like I guess maybe that's the upside, is as bad as some of these things seem, we still get a year, two, three years down the line, and we realize things are still moving.

Steve: The world did not end.

JASON: The world did not end. Although when it's happening, man, it seems like the biggest deal in the world. So at least there's that.

Steve: Well, and there is a real point there, and that is one could argue that the world did not end because we did update the servers for Heartbleed. The world did not end because we did secretly on one day announce DNS spoofability when all the servers were ready to be updated. So and the best example is the Y2K problem. I mean, I literally sat here awake, waiting to see what was going to happen because we know, if nothing had been done, a chunk of the world would have gone insane. But we knew about it in time to mitigate a huge distributed mitigation effort across the globe. And so nothing happened. And there were people who said, see, it was all a big con. It was nothing. It was a tempest in a teapot. It's like, yes, fortunately that's what we turned it into.

JASON: Exactly.

Steve: By fixing the code.

JASON: Right. But I want to see it with my own eyes. It's like, no, not good enough, even though the solution is the real story.

Steve: We've got another one coming up in 2038. That's when Unix time wraps. There is a Unix time is the number of seconds which have elapsed since some date, I don't remember when it is, but it goes back to zero in 2038. And that's not long from now.

JASON: No.

Steve: And there's a lot of systems in closets and a lot of old Unix kernels that are going to get very confused in 2038. And that's going to be tough.

JASON: I suppose we have plenty of time to signal that.

Steve: I don't know. I don't know if anybody is paying attention.

JASON: Well, yeah, exactly. It's almost too much time to signal.

Steve: Sixteen years; you know? I guess we can hope that every...

JASON: Right. Let's make that a priority right now. We'll see what happens. Well, Steve, thank you so much. Always appreciate the work that you do. I know everybody that watches and listens echoes that sentiment completely. You're a master and a pro at all things security, so we appreciate you sharing your knowledge with us every single week.

Steve: Glad to be here.

JASON: If you want to follow what Steve is doing, all you have to do is head over to his site, GRC.com. You can find all sorts of Steve goodness on the site. Of course SpinRite. You can find information about the best mass storage recovery and maintenance tool and get your copy there. You can find audio and video of this show there, as well as I believe that's the only place where you can get transcripts of this show found there, so go to GRC.com for that. We of course have a show page on the web for this show as well, TWiT.tv/sn, so you can find our audio, our video, the ways to subscribe to the podcast, jump out to YouTube. Anything you need between those two sites you're going to find it.

This show records live every Tuesday at 4:30 p.m. Eastern, 1:30 p.m. Pacific, 20:30 UTC. So if you're around at that time, and you want to watch us do the show in real-time, usually it's Leo with Steve, but today it's of course me sitting in for Leo. You can do that at TWiT.tv/live, and then of course participate in the chatroom and all that fun stuff. And finally, don't forget Club TWiT, TWiT.tv/clubtwit, that's our subscription tier for all

things we do with no ads and a whole lot of other perks and features. So TWiT.tv/clubtwit.

Thank you, Steve. Thanks for letting me crash your party this week. I always enjoy it.

Steve: Thanks for standing in for Leo, Jason. It's always a pleasure. And I think we've got, I don't know when - oh, no, I do know when. Leo does have another trip coming up. I actually have it...

JASON: He's got, well, he's got the cruise, that's right, the cruise is coming up, what is that, a month from now? A month-ish? Somewhere? I don't have the dates.