Chat (out of) Control

Description: What would you call Stuxnet on steroids? What's the latest on the Voyager 1 drama? What new features are coming to Android and Thunderbird? What's China done now? Why did Gentoo Linux say "no" to AI? And after sharing and discussing a bunch of feedback from our terrific listeners and a SpinRite update, we're going to examine the latest update to the European Union's worrisome "Chat Control" legislation, which is reportedly just over a month away from becoming law. Is the EU about to force the end of end-to-end encryption in order to enable and require the scanning of all encrypted communications? It appears ready to do just that.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-971.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-971-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. The latest chapter in the Voyager 1 drama coming up. We'll talk about the graybeard at Gentoo who says, "No AI in Linux." About the Hyundai owner whose car really is tracking him. And then what the EU plans to do with end-to-end encryption. I can give you a little tip. It's not good news. All that coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 971, recorded Tuesday, April 23rd, 2024: Chat (out of) Control.

It's time for Security Now!, the show where we cover the latest security, privacy, Internet updates, even occasionally some good books and movies, with this guy right here, Steve Gibson, the security guy in chief. Hello, security guy.

Steve Gibson: Yeah. And, you know, where are the good movies, Leo?

Leo: I know.

Steve: I mean, like, we used to have a lot of fun. I did see that you're doing the Bobiverse with Stacey's book club.

Leo: Yes, Thursday, yes.

Steve: She was chagrined when you reminded her. She said, "Ooh, I forgot to read that."

Leo: Yeah.

Steve: So of course it'll take her an hour.

Leo: It's pretty quick, yeah.

Steve: And my wife apparently just glances at pages when she - like I see her, like with eBooks, I said, "Can I test you on the content of this after you're through, like?" And she says, "Oh, I'm getting most of it. It's fine."

Leo: Did she go to Evelyn Wood's Speed Reading Academy?

Steve: Apparently. Or she's in some sort of a time warp. I don't know. Because, I mean, I'm an engineer. I read every word.

Leo: Yes.

Steve: And in fact that's why Michael McCollum was sending me his books before publication because it turns out I'm a pretty good proofreading editor because I spot every mistake. Of course not my own. So other people's. Much easier to see.

Leo: Yes. It's really true. Same with code; right? You know there's a bug in there, you can stare at it till the cows come home, and you go...

Steve: Absolutely. One of the neatest things that the GRC group has done for me, our SpinRite news group, is, you know, this thing, basically it's not - we're not having any problems. Thousands of people are coming onboard with 6.1, and it's done.

Leo: Isn't that a nice feeling?

Steve: We're not, you know, I'm not chasing bugs around.

Leo: Oh, that's nice.

Steve: You know, it is, it is good.

Leo: You've got a great bunch of beta testers.

Steve: That's in control. We're going to talk about what is out of control. Today's title is "Chat (out of) Control" for Security Now! Episode 971, for the second-to-last episode of the month of April, which becomes important because of what's going to happen in June.

But anyway, I'm getting all tangled up here. We're going to talk about a lot of fun things, like what would you call Stuxnet on steroids? What's the latest on the Voyager 1 drama? We've got even more good news than we had last week.

What new features are coming to Android, probably in 15 - we're not sure, but probably - and also in Thunderbird this summer? What's China gone and done now? Why did Gentoo Linux say "no" to AI; and, like, what's that all about? And after sharing and discussing a bunch of feedback - because there wasn't a huge bunch of, like, really gripping news, but we had a lot of feedback from our listeners that we're going to have fun with - and a brief little update on SpinRite, we're going to examine the latest update to the European Union's quite worrisome "Chat Control" legislation, which is reportedly just over a month away from becoming law.

Is the EU about to force the end of end-to-end encryption in order to enable and require the scanning of all encrypted communications for the children? And it appears ready to do just that. This latest update, it came onto my radar because somebody said that the legislators had excluded themselves from the legislation.

Leo: Of course. Of course.

Steve: Well, and so I got this 203-page tome, and its Section 16a was in bold because it just got added. Anyway, we'll talk about it. I think that the person doing that speaking that caught my attention, and I'm glad he caught my attention, but he was overstating the case in order to make a point.

Leo: Oh, okay.

Steve: But the case that we have doesn't need overstating because it looks really bad. You know, there's no sign of exclusion like the UK gave us on their legislation in September which said "where technically feasible." That's completely missing from this. So anyway, I think we have a lot of fun to talk about, fun things to talk about.

I did make sure that the pictures showed up this week in Apple devices. What's interesting is I have an older 6 - I think it's a 6, or maybe it's a 7 or 8, I don't know. Anyway, the pictures all work there. Even last week's pictures worked there. But not on my iPhone 10. So Apple did in fact change the rendering of PDFs which caused some problem, some incompatibility. Anyway, I don't know why it was last week but not this week. We're all good to go this week. So even Mac people can see our Picture of the Week. Which is kind of fun. So lots of good stuff.

Leo: I can see it, so I verify that it does in fact work. Oh, that's good news. Now I believe it's time for a Picture of the Week.

Steve: Yeah. So this just caught my attention because lately I've been seeing, as I'm sure our listeners have, so much of this. You know, AI everything.

Leo: AI, AI, AI, yup.

Steve: AI everywhere. So the picture shows a couple of young upstarts in a startup venture who are - they've got some ideas for some product that they want to create. And one of the things that happens when you're going out to seek financing and funding, you're typically going and giving presentations to, like, venture capital firms, and explaining what you're going to do and how you're going to do it.

And so PowerPoint presentations are put together, and they're called "pitch decks" because you're making a pitch to whomever you're explaining your ideas to. So we see in this picture two guys facing each other, each behind their own display, one of them saying to the other, "Can you go through all the old pitch decks and replace the word 'crypto' with 'AI'?" And of course the point being that we were just, what was it, like a year ago, Leo, that everything...

Leo: It's just a catchphrase, yeah.

Steve: Yeah, exactly. I mean, it was, like, time must be accelerating because it was just so recently that everything was blockchain this and blockchain that; crypto currency, you know, crypto this and that. And so, no, that's all yesterday. That's so, what do they say, so last minute or something? Anyway, now it's AI. So, yes. And we have a couple things during this podcast that will be touching on this, too. So anyway, just, you know, not a fantastic picture, but I thought it was just, like, so indicative of where we are today.

I've been dealing with Bing. I don't know why I've been launching it, but it's been launched a few times in the last week. And, you know, Microsoft...

Leo: Because you use Windows, that's why.

Steve: Oh, that would definitely...

Leo: They do everything they can to get Bing in your face.

Steve: Oh, my god, yes. And so it's like, no, I don't want this. And also it's, for me, since I'm not normally using Edge or Bing, it's like, okay, how do I close this? It looks like it takes over the whole UI, and very much like that old, you know, when people were being forced to upgrade to Windows 10 against their will, where for a while it said, "No, thank you," and then it changed to "Later tonight." So it's like, wait a minute. What happened to not at all, never ever? You know, it's like you want to do it now, or do you want to do it in an hour? Uh, wait. Those are my only two options?

Anyway, okay. So as we know, Security Now! is primarily an audio podcast. But even those watching, though it remains unclear to me why anyone would, don't have the advantage of looking at my show notes. If anyone were to be reading the notes, they would see that the spelling of the name of this new attack is far more, shall we say, acceptable in polite company than the attack's verbal pronunciation. But this is an audio podcast, and the story of this attack that I very much want to share refers to the attack by name. And that name, which rhymes with "Stuxnet" is spelled "F-U-X-N-E-T." And there's really no other way to pronounce it than just to spit it out. But I'm just going to say Fnet for the sake of the children.

Leo: Thank you, Steve. Thank you.

Steve: Because, yes, you know. So it's not really an F-bomb. But it's audibly identical, and there's no point in saying it. Everybody understands how you would pronounce F-U-X-N-E-T. Which is what the Ukrainians named the weapon which they reportedly, and this was confirmed by an independent security company, successfully launched into the heart of Russia.

So with that preamble and explanation, let's look at the very interesting attack that was reported last week by Security Week. Their headline, which also did not shy away from using the attack's name, said "Destructive ICS Malware 'Fuxnet' used by Ukraine Against Russian Infrastructure." So here's what we learn from what they wrote. They said: "In recent months, a hacker group named Blackjack, which is believed to be affiliated with Ukraine's security services" - so, you know, as in state-sponsored - "has claimed to have launched attacks against several key Russian organizations.

"The hackers targeted ISPs, utilities, data centers, and Russia's military, and allegedly caused significant damage and exfiltrated sensitive information. Last week, Blackjack disclosed the details of an alleged attack aimed at Moscollector (M-O-S-C-O-L-L-E-C-T-O-R), a Moscow-based company responsible for underground infrastructure, meaning things like water, sewage and communication systems."

So, quoting, they said: "Russia's industrial sensor and monitoring infrastructure has been disabled." So said the hackers. "It includes Russia's Network Operation Center that monitors and controls gas, water, fire alarms, and many others, including a vast network of remote sensors and IoT controllers.

"So the hackers claimed to have wiped database, email, internal monitoring, and data storage servers. In addition, they claimed to have disabled some 87,000" - 87,000 - "sensors, including ones associated with airports, subway systems, and gas pipelines. To achieve this, they claimed to have used Fuxnet, a malware they described as 'Stuxnet on steroids,' which enabled them to physically destroy sensor equipment."

You know, our longtime listeners and anybody's who's been in, you know, around IT will recall that Stuxnet was a previous, also physically destructive malware. I guess we have to call it malware, even though we were apparently part of - the U.S. participated, or U.S. intelligence services was involved in its creation. It caused the centrifuges used in Iran to overspin and essentially self-destruct. So those were being used to enrich uranium at the time. Anyway, so that's why they're calling this thing "Stuxnet on steroids" is that they worked to cause actual physical damage, as we'll see in a second, to hardware.

Leo: There's a big difference, though, between destroying centrifuges which have one purpose, which is enriching uranium, and destroying sensors which prevent gas leaks and, I mean...

Steve: Yeah, yeah.

Leo: I mean, this is a civilian attack. Finish the story, but I would love to talk at the end of it about how you feel about this.

Steve: Good. And I agree with you. So they wrote: "Fuxnet has now started to flood the RS485/MBUS and is sending random commands to 87,000 embedded control and sensory systems." And they did say "(while carefully excluding hospitals, airports, and other civilian targets)." Now, they said that. So, you know, they share some of our sensitivity to that. And I do question, you know, given that they're also claiming 87,000-some sensors, how they can be that careful about what's, you know, what they've attacked and what they haven't.

Anyway, the report goes on, saying: "The hackers' claims are difficult to verify, but the industrial and enterprise IoT cybersecurity firm Claroty was able to conduct an analysis of the Fuxnet malware based on information and code made available by Blackjack. Claroty pointed out that the actual sensors deployed by Moscollector, which are used to collect physical data such as temperature, were likely not themselves damaged by Fuxnet. Instead, the malware likely targeted roughly 500 sensor gateways."

So, right? So the idea is that the gateway is a device out located remotely somewhere, and it has RS485 lines running out to a ton of individual sensors. So it's the sensor data collector and forwarding device. So the malware targeted around 500 of these sensor gateways, which communicate with the sensors over a serial bus such as RS485 or Meter-Bus that was mentioned by Blackjack. These gateways are also connected to the Internet to be able to transmit data to the company's global monitoring system. So that was probably the means by which the Fuxnet malware got into the sensor gateways.

"Claroty notes: 'If the gateways were indeed damaged, the repairs could be extensive given that these devices are spread out geographically across Moscow and its suburbs, and must be either replaced or their firmware must be individually reflashed.' Claroty's analysis of Fuxnet showed that the malware was likely deployed remotely. Then, once on a device, it would start deleting important files and directories, shutting down remote access services to prevent remote restoration, and deleting routing table information to prevent communication with other devices.

"Fuxnet would then delete the file system and rewrite the device's flash memory. Once it has corrupted the file system and blocked access to the device, the malware attempts to physically destroy the NAND memory chip and then rewrites the UBI volume to prevent rebooting.

"In addition, the malware attempts to disrupt the sensors connected to the gateway by flooding their serial communications channels with random data in an effort to overload the serial bus and sensors, essentially performing an internal DoS attack on all the devices the gateway is connected to." And I'll argue that if these are not sensors, but these are actuators, as you said, Leo, this could be causing some true damage. I mean, like true infrastructure [crosstalk] damage.

Leo: Well, they said subway systems, airports, gas pipelines.

Steve: Yeah. "Claroty explained: 'During the malware operation, it will repeatedly write arbitrary data over the Meter-Bus channel. This will prevent the sensors and the sensor gateway from sending and receiving data, rendering the sensor data acquisition useless. Therefore, despite the attackers' claim of physically destroying 87,000 devices,' wrote Claroty, it seems that they actually managed to infect the sensor gateways and were causing widespread disruption by flooding the Meter-Bus channel connecting the sensors to the gateway, similar to network fuzzing the different connected sensor equipment. As a result, it appears only the sensor gateways were bricked, and not the end-sensors themselves.'"

So, okay. I particularly appreciated the part about attempting to physically destroy the gateway's NAND memory chip. Because it could happen. As we know, NAND memory is fatigued by writing because writing and erasing, which needs to be part of writing, is performed by forcing electrons to tunnel through insulation, thus weakening its dielectric properties over time. So the attacking malware is likely writing and erasing and writing and erasing the NAND memory over and over, as rapidly as it can. And since such memory is likely embedded into the controller and is probably not field replaceable, that would necessitate replacing the gateway device, and perhaps all 500 of them spread across Moscow and its suburbs.

And even if the NAND memory was not rendered unusable, the level of destruction appears to be quite severe. Wiping stored data and directories and killing the system's boot volume means that those devices probably cannot be remotely repaired. Overall, I'd have to say that this extremely destructive malware was well named.

And we live in an extremely, and increasingly, cyber-dependent world. Everyone listening to this podcast knows how rickety the world's cybersecurity truly is. So I shudder at the idea of any sort of all-out confrontation between super powers. I don't want to see that.

Leo: Do you think there should be a, I don't know, Geneva Convention-style accord between nations about cyberwarfare? I mean, it's - the problem is you can do it, but then you're just going to escalate. It's going to go back and forth. Which is why we decided, for instance, not to allow bioweapons. Now, they still get used. But it's, you know, the civilized world agrees not to use biologic weapons in war.

Steve: Well, and the feeling is, of course, that COVID was a lab escape. Right?

Leo: Well...

Steve: I mean...

Leo: There's some evidence, but not a lot.

Steve: There's no evidence.

Leo: That's a question, yeah. It wasn't a very good, it wasn't a very effective warlike attempt since it killed far more people in China than it did elsewhere. But anyway...

Steve: It was clearly a mistake.

Leo: Yes. It wasn't intentional. So what do you think? I mean...

Steve: So I agree with you. The problem is it's tempting because it doesn't directly hurt people; right? I mean, so like right now we're in a cold war. We're constantly on this podcast talking about state-sponsored attacks. Well, those are attacks.

Leo: Yeah. And especially infrastructure attacks.

Steve: Yes.

Leo: Which this was.

Steve: Yes. I mean, the whole Colonial Pipeline thing; you know?

Leo: Right.

Steve: That really damaged the U.S. And, I mean, it was a true attack. So, you know, and we just talked about how China was telling some of their - China told their commercial sector, you need to stop using Windows. You need to stop using, you know, this Western computer technology because the West is able to get into it. So that was the first indication we really had that, as I put it at the time, that we're giving as well as we're getting. Unfortunately, this is all happening. I mean, I wish none of it was happening. But the problem is security is porous. And I guess the reason a nuclear weapon and a bioweapon are unconscionable, you know, is that they are so tissue damaging, for lack of a better word. I mean, they really - they're, like, really going to kill people.

Whereas, eh, a network got breached, whoops. You know, I mean, it doesn't have the same sort of visceral grip. And unfortunately, here's an example. And I'm glad you brought it up, Leo. Ukraine, sympathetic as we can be for their situation, this was a blunt-edged attack; right? I mean, this was, you know, sewage and water and gas and airports and, you know, I mean, it's - they couldn't have controlled what damage was caused. And, you know, you mess up water and sewage, and you're really hurting actual people who are innocent of what, you know...

Leo: Or subways. Or airports. Or gas pipelines. I don't know what the answer is. I mean, I'm no fan of Putin. He brought the war upon himself. But hurting civilians, I don't know, this is not a good situation.

Steve: It's the world we're in.

Leo: It's the world we're in.

Steve: Yeah. And it is technology we created. I mean, you know, oh, let's have the password be admin/admin because we don't want people calling us and asking what the password is. Or, I mean, it's like we've made so many bad decisions. And while we're now making them better today, we have seen how long the tail of inertia is. I mean, it's, you could argue, infinite. You know? We still have Code Red and Nimda out there, you know, sending packets out. Somewhere there's an NT machine just hoping to find something that it can infect. When is it going to die? I don't know.

We have another update on Voyager 1. Apparently, if Voyager is not going to give up on us, we're not going to give up on it. But remember that, no matter what, Voyager is deriving all of its diminishing operating power from the heat being generated by the

decay of radioisotopes. And through the years and now decades, since this thing left Earth in '73, those isotopes are continuing to put out less and less heat. And thus Voyager has less and less energy available to it. So it can't go forever. But it, you know, it amazes everybody that it's gone as long as it has, and it is still going.

What equally amazes me is the intrepid group of well-past-their-retirement engineers who are now endeavoring to patch the code of this ancient machine that's 22.5 light hours away.

Leo: Oh, my god. It's amazing.

Steve: It boggles the mind.

Leo: It's so amazing.

Steve: Just yesterday, on April 22nd, JPL, NASA's Jet Propulsion Laboratory, posted the news under the headline "NASA's Voyager 1 Resumes Sending Engineering Updates to Earth." They wrote: "After some inventive sleuthing, the mission team can for the first time in five months check the health and status of the most distant human-made object in existence. For the first time since November, NASA's Voyager 1 spacecraft is returning usable data about the health and status of its onboard engineering systems.

"The next step is to enable the spacecraft to begin returning science data again. The probe and its twin, Voyager 2, are the only spacecraft to ever fly in interstellar space, the space between the stars. Voyager 1 stopped sending readable science and engineering data back to Earth on November 14th, 2023, even though mission controllers could tell the spacecraft was still receiving their commands and otherwise operating normally. In March, so last month, the Voyager engineering team at NASA's Jet Propulsion Laboratory in Southern California confirmed that the issue was tied to one of the spacecraft's three onboard computers, called the flight data subsystem (FDS). The FDS is responsible for packaging the science and engineering data before it's sent to Earth.

"The team discovered that a single chip responsible for storing a portion of the FDS's memory including some of the FDS computer's software code is no longer working. The loss of that code rendered the science and engineering data unusable. Unable to repair the chip" - right, after 22.5 light, what is it, days, light days away - "the team decided to place the affected code elsewhere." They're relocating the code, Leo, at this distance.

Leo: I know, it's amazing.

Steve: On a probe built in, or launched in '73.

Leo: How cool.

Steve: It's, you know, it's insane. "But," they said, "no single location is large enough to hold the section of code in its entirety." So they're having to fragment it.

"They devised a plan to divide the affected code into sections and store those sections in different places in the FDS. To make this plan work, they also needed to adjust those

code sections to ensure, for example, that they all still function as a whole. Any references to the location of that code in other parts of the FDS memory need to be updated, as well." So they're relocating and then patching to relink the now-fragmented code sections so that they jump to each other. It's, you know, dynamic linking in a way that was never designed or intended.

They wrote: "The team started by singling out the code responsible for packaging the spacecraft's engineering data. They sent it to its new location in the FDS memory on April 18th. A radio signal takes about 22.5 hours to reach Voyager 1, which is now over 15 billion" - with a "b" - "miles from Earth, and another 22.5 hours" - hours, not days - "for a signal to come back to Earth. When the mission flight team heard back from the spacecraft on April 20th, they saw that the modification worked. For the first time in five months, they have been able to check the health and status of the spacecraft.

"During the coming weeks, the team will relocate and adjust the other affected portions of the FDS software. These include the portions that will start returning science data," rendering the satellite again back to doing what it was designed to do, which is using its various sensor suites and sending back what it's seeing and finding out in interstellar space, which as I mentioned previously has surprised the cosmologists because their models were wrong. So Voyager 1 is saying, uh, not so fast there. Nice theory you've got, but it's not matching the facts. Wow.

Leo: Yay, V'Ger.

Steve: Yeah.

Leo: And yay, those brilliant scientists who are keeping her alive.

Steve: Oh, and Leo, I did take - Lorrie and I watched "It's Quieter in the Twilight."

Leo: Oh, yeah, "It's Quieter in the Twilight," yeah.

Steve: And what was interesting was that this announcement, and it was picked up in a few other outlets, showed a photo of the event where the team were gathered around their conference table. I recognized them.

Leo: Yes.

Steve: From the documentary.

Leo: It's the same people since 1974.

Steve: Yeah, exactly. They're all still there. And in fact some of them don't look like they've changed their clothes, but that's what you get, you know, with old JPL engineers.

Leo: Oh, I just love it. It's such a great, wonderful story. It really is.

Steve: Let's take a break. I'm going to catch my breath, and then we're going to talk about changes coming to Android 15 and Thunderbird.

Leo: Yes. All right. As we continue with the best show on the podcast universe, 22 light hours ahead of everyone else. On we go.

Steve: Okay. So there's not a lot of clear information about this yet, but Google is working on a new feature for Android which is interesting. They're going to start watching their apps' behavior. It will place under quarantine any applications that might sneak past its Play Store screening, only to then begin exhibiting signs of behavior that it deems to be malicious. The apps will reportedly have all their activity stopped, all of their windows hidden, and notifications from the quarantined apps will no longer be shown. They also won't be able to offer any API-level services to other apps.

The reports are that Google began working on this feature during Android 14's development last year, and that the feature is expected to finally appear in forthcoming Android 15. But we don't have that confirmed for sure. So, you know, there wasn't - I wasn't able to find any dialogue or conjecture about why the apps aren't just removed. Maybe - oh, and that they do still appear to be an app installed on the phone. They're not hiding it from the user. They're just saying, no, you bad app, we don't like what you've been doing.

Maybe it reports back to the Play Store, and then Google takes a closer look at the app which is in the Play Store, which of course is how the user got it, and then says, oh, yeah, we did miss this one. And at that point it gets yanked from the Play Store and yanked from all the Android devices. So it could just be like essentially functioning as a remote sensor package. Anyway, I'm sure we'll learn more once it becomes official, hopefully in this next Android 15.

Also this summer, Thunderbird will be acquiring support for Microsoft Exchange email for the first time ever. It will only be email at first. The other Exchange features of Calendar and Contacts are expected to follow at some later date, although Mozilla's not saying for sure. Now, I happen to be a Thunderbird user. I was finally forced to relinquish the use of my beloved Eudora email client once I began receiving email containing extended non-ASCII character sets that Eudora was unable to manage. I got these weird capital A with little circles above them things in my email, which was annoying, instead of line separators.

At the same time I have zero interest in Exchange. GRC runs a simple and straightforward instance of a mail server called hMailServer which handles traditional POP, IMAP, and SMTP, and does it effortlessly with ample features. But I know that Exchange is a big deal, and obviously Mozilla feels that for Thunderbird to stay relevant, it probably needs to add support for Exchange.

In any event, to support this rather massive coding effort, in Mozilla's reporting of this they mentioned that it had been 20 years since - because, you know, email is kind of done - 20 years since any code in Thunderbird dealing with email had been touched. They've just been, you know, screwing around with the user interface. And that during those 20 years a lot of, as they put it, institutional knowledge about that code had drained. So they've decided now that they're going to recode in Rust. Rust is their chosen implementation language. And they did so for all the usual reasons.

They cited memory safety. They said: "Thunderbird takes input from anyone who sends an email, so we need to be diligent about keeping security bugs away." Performance:

"Rust runs as native code with all the associated performance benefits." And modularity and ecosystem. They said: "The built-in modularity of Rust gives us access to a large ecosystem where there are already a lot of people doing things related to email which we can benefit from," they said. So anyway, for what it's worth, you know, Thunderbird is a strong client for, you know, from Mozilla. Is it multiplatform, Leo, do you know? Is Thunderbird Windows-only? Or Mac and Linux? I don't know either way.

Anyway, China. The Chinese government has ordered Apple to remove four Western apps from the Chinese version of the Apple App Store. Those are Meta's new social network Threads, which is now gone; Signal; Telegram; and WhatsApp, all removed from the Chinese App Store. China stated that they have national security concerns about those four. And as we've seen, and as I fear we'll be seeing shortly within the EU, what countries request, countries receive. Technology is ultimately unable to stand up to legislation. And this is going to cause a lot of trouble, as I mentioned, in the EU. We'll be talking about that here at the end of the podcast.

Leo: Yeah. And I think the Chinese government's removing it for the same reason the EU wants to remove it. They don't like end-to-end encryption.

Steve: Yes, exactly.

Leo: Threads is something else. But Signal and Telegram and WhatsApp, that's all E2E encryption.

Steve: Yeah.

Leo: By the way, to answer your question, I was down the hall. Thunderbird is Mac, Windows, Linux.

Steve: Okay.

Leo: It's completely open source and everywhere.

Steve: Very cool.

Leo: Nice program.

Steve: In that case, it will have access to Exchange Server, which may allow it to move into a corporate environment, which is probably what they're thinking.

Leo: Which would be great, yes.

Steve: Yeah.

Leo: That would be great, yeah. Well, we'll see if Microsoft does it. Oh, no, they're going to do it. Oh, cool.

Steve: Oh, yeah. You mean Mozilla.

Leo: I just wish they'd kill Exchange Server.

Steve: And, I mean, just get out of the Exchange Server, like...

Leo: I wish Microsoft would kill Exchange. That's been a problem since forever.

Steve: Yeah, yeah. Since it was created.

Leo: Exactly, yeah.

Steve: And do you think that China's move is like in response to TikTok and what's happening here in the U.S.?

Leo: Well, I mean, we had a discussion on MacBreak about that. The Times says it's because nasty things were said on those platforms about Xi Jinping. Which is possible. There's no corroboration of that. Apple says no. I think it's just that they were - Threads, maybe that would be because Threads doesn't have any, you know, it's just a social network. But for sure they don't want - I think Threads is being killed probably because of TikTok. Andy I think pointed out that it happened immediately after the TikTok ban was approved in the House.

Steve: Ah.

Leo: So it's likely, by the way, that that this time will be approved in the Senate because it's part of a foreign aid package.

Steve: Right.

Leo: So get ready to say goodbye to TikTok.

Steve: Wow, Leo. That'll be an event, won't it.

Leo: I don't think - I think the courts will block it. I'm hoping they will, but I don't know. It's a very weird thing. They have a year and a half to do it.

Steve: Well, and, I mean, here, you know, we were talking about a cold war. And there's this, you know...

Leo: There's an economic cold war, absolutely, yeah.

Steve: Right. And China understandably is uncomfortable about Western-based apps using encryption that they're unable to compromise.

Leo: Right, right.

Steve: So, I mean, I get it. You know? And so it's sort of like we lived through this brief period where, you know, there was global encryption and privacy, and everybody had apps that everybody could use. And then barriers began getting erected. Right? I mean, so, sorry. If you're Chinese, you've got to use China Chat.

Leo: Well, and the numbers for these particular apps in China are pretty low. We're talking hundreds of thousands of users, not millions.

Steve: Ah, okay.

Leo: Or billions, yeah.

Steve: Okay. So not a huge actual impact.

Leo: I think it's an easy thing for them to do, yeah.

Steve: Okay. So this was interesting. I'll just jump right in by sharing the posting to the Gentoo mailserve. This was posted by a longstanding, since 2010, so 14 years of involvement, and he's very active, Gentoo developer and contributor. He wrote: "Given the recent spread of the 'AI' bubble" - and he has AI in quotes, like, everywhere, so he's obviously immediately exposed himself as not being a fan.

Leo: Gentoo, you should understand, is the ultimate graybeard Linux; okay? That's all you need to know.

Steve: That totally explains it, yes. "Given the recent spread of the 'AI' bubble, I think we really need to look into formally addressing the related concerns." Oh. He says: "In my opinion, at this point the only reasonable course of action would be to safely ban 'AI'-backed contribution entirely. In other words, explicitly forbid people from using ChatGPT, Bard, GitHub Copilot and so on, to create ebuilds, code, documentation, messages, bug reports and so on for use in Gentoo. Just to be clear, I'm talking about our original content. We can't do much about upstream projects using it."

Then he says: "Here's the rationale: One, copyright concerns. At this point, the copyright situation around generated content is still unclear. What's pretty clear is that pretty much all LLMs (Large Language Models) are trained on huge corpora of copyrighted material, and the fancy 'AI' companies don't care about copyright violations. What this means is that there's good risk that these tools would yield stuff we cannot legally use. Two,

quality concerns. LLMs are really great at generating plausible-looking BS." And he didn't actually say "BS," but I changed it for the podcast.

Leo: He's cranky.

Steve: Oh, he is.

Leo: That's what happens when you have to compile every piece of software you use from scratch. It makes you cranky.

Steve: Right.

Leo: It makes you cranky.

Steve: That's right. "I suppose," he says, "they can provide good assistance if you are careful enough, but we can't really rely on all our contributors being aware of the risks." Then there's ethical concerns, number three. "As pointed out above, the 'AI' corporations care about neither copyright nor people."

Leo: That's probably true.

Steve: Yeah. "The 'AI' bubble is causing huge energy waste. It's giving a great excuse for layoffs and increasing exploitation of IT workers. It is driving the further" - and here I felt I had to use the word because it is now become a common word. And I think I've heard it on the TWiT Network.

Leo: Yeah, we allow this word.

Steve: "The 'enshittification' of the Internet. It is empowering all kinds of spam and scam." And that is the case. "Gentoo has always stood," he concludes, "as something different, something that worked for people for whom mainstream distros were lacking. I think adding 'made by real people' to the list of our advantages would be a good thing. But we need to have policies in place to make sure that AI-generated crap" - and again, not the word he chose - "doesn't flow in."

Leo: I like this guy. He's right. I think that's fair. Did you see the study from the University of Illinois in Urbana-Champaign?

Steve: No.

Leo: They used ChatGPT 4, the latest version of OpenAI's model. They gave it the CVE database. That's it. Nothing more than the description and the CVE. And it was able to successfully attack 87% of those vulnerabilities. It was able to craft an attack based merely on the CVE description, an effective attack.

Steve: Wow. Wow.

Leo: Yeah, I mean, I think he's probably right. But I don't know how you enforce this because...

Steve: No. That's exactly the problem.

Leo: Yeah.

Steve: In his posting, he had a link, he referred to some, we'll call it a "crap storm," over on GitHub. And I went, I followed the link because I was curious. There is a problem underway where what is clearly AI-generated content which, you know, looks really reasonable, but doesn't actually manage to get around to saying anything, is like becoming a problem over on GitHub.

Leo: Yeah.

Steve: So anyway, in order to share that, as we saw, I had to clean up the language in his posting since he clearly doesn't think much of AI-generated code. And as I said, there have been some signs over on GitHub, which he referred to, of descriptions appearing to be purely AI-generated. You know, they're not high quality. And I suppose we should not be surprised, Leo, that there are people, maybe we'll call them "script kiddies," who are probably incapable of coding from scratch for themselves. So why wouldn't they jump onto Large Language Model systems, which would allow them to feel as though they're contributing. But are they really contributing?

Leo: Now, look, let's face it. Humans are just as capable of introducing bugs into code as anybody else, and more often maliciously than AIs. I mean, AIs aren't natively malicious. The other thing I would say is there is a lot of AI-generated prose on GitHub because English is often not the first language of the people doing the coding. A lot of repositories on GitHub are by non-English speakers. And I think that that's more likely the reason you'll see kind of AI-like prose on there because they don't speak English that well, or at all. And so they're using ChatGPT, for instance, to generate the text.

Steve: Right.

Leo: Human, I don't think, I mean, honestly, I've used Copilot. I have my own custom GPT for LISP. The code it generates is indistinguishable from human code, probably because it is, at some point, from human code.

Steve: Right, right.

Leo: I don't know how you're going to stop it. It's not - doesn't have a big red flag that says an AI generated this.

Steve: Right. And as we've noted, the genie is out of the bottle already. So, yeah. We are, we're definitely in for some interesting times.

Okay. We've got a bunch of feedback that I found interesting, that I thought our listeners would, too. Let's take another break, and then we will get into what - what was this one? Oh, oh. We have a listener whose auto was spying on them, and he's absolutely sure it never had permission. And we have a picture of the report that it generated.

Leo: We here at Nissan see that you've been using your vehicle for lovemaking. And we want you to knock it off. But we'll find out more about that in just a second. But first, a word.

Steve: And apparently you're doing it wrong.

Leo: You're doing it wrong. We have some tips we'd like to share. Steve, let's close the loop.

Steve: So we have a note from a guy who is no slouch. He's a self-described user of ShieldsUP!, SpinRite, and an avid listener of Security Now!. He's also an Information Security Practitioner, and I think he said a Computer Geek. Oh, yeah, he does. So he said: "Hi, Steve. I apologize for sending to this email" - probably came through Sue or Greg - "as I couldn't find a different email for contact information." And yes, that's by design, but okay. He said: "Anyway, longtime follower of ShieldsUP! and SpinRite, and an avid listener of Security Now!. My full-time gig is an InfoSec Security Practitioner/ Computer Geek.

"We have a couple of Hyundais in the family, and I purchased one last fall. I use the Hyundai Bluelink app on my phone, as I can make sure I locked my doors, and get maintenance reminders. I made a point to NOT opt-in for the [he has in quotes] 'driver discount'; and, as a privacy cautious person, I decline sharing data wherever possible. But after the story in The New York Times regarding carmakers sharing data, I contacted Verisk and LexisNexis to see what they had on me.

"LexisNexis had nothing other than the vehicles I have owned in the past, but Verisk had a lot. I have attached a page of the report. It includes driving dates, minutes (day and night), acceleration events, and braking events. The only thing missing is the actual speeds I was going, or if I was ever speeding.

"What bothers me most about this is that I have no way to challenge the accuracy. For events that are not illegal, I can still be penalized. Braking hard and accelerating fast should not be safety concerns without context. And today's smarter cars are still imperfect. My adaptive cruise control" - he has in parens (radar) - "will still brake hard at times it shouldn't, and I will get penalized by that data. My car is also a turbo, and if I accelerate for fun or safety, that too can be a penalty. And if I happen to drive in Texas where there are highways with an 85 MPH speed limit, I would be down-rated for that legal behavior.

"My family tried the 'safe driving' BT dongles from another insurer years ago, but the app had too many false positives for driving over speed" - he said posted speed limit doesn't agree with the app - and hard braking and accelerating, that we decided it wasn't worth our time or the privacy concerns. My wife and I are close to Leo's age, and she drives like a grandmother, but her scores were no better than mine.

"I have attached a picture of the document I got from Verisk," he says, "name and VIN Number removed, to give you an idea of what is reported without my consent from my car. I've contacted Hyundai and told them I do not and did not consent to them sharing my data with Verisk. After a few back and forths, I got this reply on April 12th: 'Thank you for contacting Hyundai Customer Care about your concerns. As a confirmation, we have been notified today that the driver's score feature and all data-collecting software has permanently disabled. We do care. As always, if you ever need additional assistance, you can do so either by email or phone. Case number....'"

So he said: "I will request another report from Verisk in the future to validate this report from Hyundai. Keep up the good work. I thought you would like to see the data and hear from someone who is 100% certain they never opted in. All the best, Andrew."

And in the show notes, sure enough, we've got the page with a report from the period September 26th of last year through March 25th of this year. So just last - toward the end of last month, showing things like the number of trips, vehicle ignition on to ignition off was 242 instances. Speeding events where the vehicle speed is greater than 80 miles per hour has an NA. Hard braking events, where they say change in speed is less than, because it's braking, negative 9.5 kph per second is 24. So during that period of time what the car regarded as a hard braking event occurred 24 times.

Rapid acceleration events, change in speed is greater than 9.5 kph per second is 26. Daytime driving minutes between the hours of 5:00 a.m. and 11:00 p.m., 6,223. Nighttime minutes, actually very few between 11:00 p.m. and 5:00 a.m., just 25 minutes. Miles driven, 5,167.6 miles during this period. And then an itemized daily driving log showing the date, the number of trips taken that day, the number of speeding events, the number of hard braking events, rapid acceleration events, driving minutes both daytime and nighttime.

So, yes, just to close the loop on this, as we first talked about from The New York Times reporting which informed us that both this Verisk and LexisNexis were selling data to insurers, and as a consequence those insurers were relying on that data to set insurance premium rates.

Leo: And look what it says at the bottom: "This report may display driving data associated with other individuals that operated insured's vehicle."

Steve: Yup.

Leo: So my guess, this is a report for an insurance company; right?

Steve: Right, right.

Leo: Whether he agreed to it or not, it may be that he can turn off some things, like I noticed that speeding events is N/A all the way through. Either he's a really careful driver, or they're not recording that. Which may well be something he didn't agree to; right?

Steve: Yup. So anyway...

Leo: I know my BMW records that because I have it on my app. And my Mustang used to give me a report card after every trip.

Steve: Right. And, I mean, compared to the way I used to drive when I was in my younger years...

Leo: Yeah, good information, yeah.

Steve: I would be happy to have my insurance company privy to the fact that I drive about three miles a day at 60 miles an hour, surrounded by other traffic. I mean, it's just, you know.

Leo: Here's my driving performance for the month of March.

Steve: And in fact Lorrie added me to her car insurance, and her rate went down.

Leo: Yes, exactly. Because you're safe; right?

Steve: Yeah.

Leo: This is more because it's an EV. You want to know a little bit about - reason you want to know about hard braking and hard acceleration and stuff is going to [crosstalk].

Steve: Right, battery treatment and, yeah.

Leo: Battery, right, right. So I think that's - I think that's great. You know. But, yeah, I understand why he doesn't want Hyundai to record it.

Steve: Well, and I would argue that a consumer who says no, I don't want to be watched and spied on and reported on, that ought to be a privacy right is available.

Leo: Yeah.

Steve: Yeah.

Leo: I'd like to see the fine print in the rest of the contract. Those are long, those contracts, you know. They go on and on.

Steve: Yeah. So Lon Seidman said: "I'm listening to the latest Security Now! episode. Definitely agree that freezing one's credit needs to be the default position these days."

Leo: Yes, yes.

Steve: "One question, though. Most of these credit agencies rely on the types of personal information that typically get stolen in a data breach for authentication. Certainly a bad actor will go for the lowest hanging fruit and perhaps move on from a frozen account. But if there's a big whale out there, they may go through the process of unlocking that person's credit, then stealing their money. What kind of authentication changes do you think are needed?"

Okay, well, that's an interesting question. Since I froze my credit reporting, I've only had one occasion to temporarily unfreeze it, which is when I decided to switch to using an Amazon credit card for the additional purchase benefits that it brought since I'm a heavy Amazon user. And that's when I discovered to my delight that it was also possible to now specify an automatic refreeze on a timer...

Leo: Yes. Isn't that great?

Steve: ...to prevent the thaw from being inadvertently permanent. Since I had very carefully recorded and stored my previously freezing authentication, I didn't need to take any account recovery measures. So I can't speak from experience. But one thing does occur to me is that strong measures are available. The reporting agencies, for example, will have our current home address. So they could use the postal system to send an authentication code via old school paper mail that would be quite difficult, if not effectively impossible, for a criminal located in some hostile foreign country to obtain. So there certainly are strong authentication measures that could be employed if needed.

Again, I don't have any experience with saying, whoops, I forgot what you told me not to forget when I froze my credit. So, but it's me. Hi. It's me, really. Unfreeze me, please. You know, because Lon's right that, you know, so much information is in the report, or in the data which is being leaked these days, for example in that massive AT&T leakage, that, you know, something over and above that needs to be used.

Leo: They gave me a long PIN. I mean, like a really long PIN.

Steve: Yes. I have that, too. I'm not, you know, I wrote it down and saved it.

Leo: Yeah. But...

Steve: But then what happens if you say, oh, it's me.

Leo: I forgot, yeah.

Steve: Yeah.

Leo: They say if you don't, you know, you can't log in, you forget it, call us. Which means you could easily social engineer a customer service rep because, let's face it, the credit reporting agencies don't want you to have a credit freeze.

Steve: Correct. Correct.

Leo: That's how they make their money is selling your information.

Steve: Correct.

Leo: So I suspect it's pretty easy to get it turned off. I would guess. By a third party.

Steve: Eric Berry, he tweeted: "What was that credit link from the podcast? I tried the address you gave out and got 'page not found.'"

I don't know why. But it is grc.sc/credit. And I just tried the link, and it works. So grc.sc/credit. That bounces you over to the Investopedia site. And I've just verified it. I said that it is still working. And for what it's worth, in what, page 9 of the show notes is the Investopedia link, all the way spelled out. So if something about your computer doesn't follow HTTP 301 redirects, then the link is there. And it's at Investopedia, How to Freeze and Unfreeze Your Credit. So you could probably also just google that.

Leo: I should also point out the FTC has a really good page about credit freezes, fraud alerts, what they are, how they work and so forth.

Steve: Yeah.

Leo: Yeah. You could also just google "FTC and credit freeze," and they have a lot of information on there.

Steve: Does that provide links to the actual freezing pages in the bureaus?

Leo: Yeah.

Steve: Because that's why I chose - oh, good.

Leo: Yeah. Absolutely.

Steve: Good, good, good. Okay.

Leo: They point you to a website run by the FTC called IdentityTheft.gov. And they're going to give you those three. Now, I should point out, there's more than three. These are the three big ones. But when I did a credit freeze, I think I did five or six of them. There's others. And it's probably not a bad idea to seek them all out. But obviously these are the three the FTC mentions, as well as Investopedia.

Steve: Right. So someone who tweets from the handle or the moniker The Monster, he said: "@SGgrc The race condition isn't solved solely with the exchange 'counter ownership' protocol unless the owner immediately rereads the owned memory region to be sure it wasn't altered before it got ownership."

Okay, now, I don't think that's correct. There are aspects of computer science that are absolutely abstract and purely conceptual. And I suppose that's one of the reasons I'm so drawn to it.

Leo: Yeah, me, too.

Steve: I think, Leo, I think you are, too.

Leo: Yes, exactly.

Steve: One of the time-honored masters of this craft is Donald Knuth. And the title of his masterwork, it's a multivolume, just it's a tour de force.

Leo: I have all three, although there supposedly are five. He's working on them.

Steve: Yes. He calls the other ones "fascicles." And I have those, as well.

Leo: Oh, yeah? How do you get those?

Steve: Oh, yeah, they're available.

Leo: I wanted the full bookshelf, but I only could find three.

Steve: Yeah. Three are in that original nice classic binding.

Leo: Yeah, beautiful, yeah.

Steve: And then he has a set of what he calls "fascicles."

Leo: Okay.

Steve: Which are the other two.

Leo: "The Art of Computer Programming." They're brilliant, brilliant, yes.

Steve: Yes, it's titled, the masterwork is "The Art of Computer Programming." And saying that is not hyperbole.

Leo: Yeah, yeah.

Steve: There are aspects of computer programming that can be true art, and his work is full of lovely constructions similar to the use of a single exchange instruction being used to manage inter-thread synchronization. In this case, as I tried to carefully explain last week, the whole point of using a single exchange instruction is that it is not necessary to reread anything because the act of attempting to acquire the ownership variable acquires it only if it wasn't previously owned by someone else, while simultaneously - and here simultaneity is the point and the requirement - also returning information about whether the variable was or was not previously owned by any other thread.

So if anyone wishes to give their brain a bit more exercise, think about the fact that in an environment where individual threads of execution may be preempted at any instant, nothing conclusive can ever be determined by reading the present state of the object ownership variable. Since the reading thread might be preempted immediately following that reading, and during its preemption the owner variable might change, the only thing that anyone reading that variable might learn, that is, just simply reading it, is that the object being managed was or was not owned at the instant in time of that reading.

While that might be of some interest, it's not interesting to anyone who wishes to obtain ownership, since that information was already obsolete the instant it was obtained. So that's what is so uniquely cool about this use of an exchange instruction which both acquires ownership only if it isn't owned and returns the previous state, meaning if it wasn't previously owned, now the thread that asked owns it. And it's as simple as a single instruction, which is just, you know, conceptually so cool.

Javamantis said: "Regarding episodes 970 and 969 with 'push button hardware config' options, my first thought is of the 2017 Saudi chemical plant attacked with the Triton malware. The admins working on the ICS controllers deliberately left an admin permission key in the controllers, instead of walking the 10 minutes required to insert the key every time a configuration needed changing."

Leo: I don't blame them. That's 10 minutes, man.

Steve: Uh-huh. "As a result, the attackers were able to access the IT systems and then the OT systems because the key was always left in and in admin mode." He says: "Lazy people will always work around inconvenient, very secure systems."

Leo: Like me, yes.

Steve: And he finishes with "To 999 and beyond, like Voyager." Yes, this podcast is going into interstellar space.

Leo: 999 and beyond. To boldly go where no podcast has gone before.

Steve: That's not true, though, because I keep hearing TWiT talking about, oh, we - yeah, that's right. Anyway, I thought he made a good point. For example, the push button dangerous config change enabler should work on a change from not pushed to pushed, rather than on whether the button is depressed. The electrical engineers among us will be familiar with the concept of "edge triggered" versus "level triggered." If it's not done that way, people will simply depress the button once, then do something like wedge a toothpick into the button in order to keep it depressed.

My feeling is, the ability to bypass well-designed and well-intentioned security does not matter at all. There's a huge gulf separating "secure by design" and "insecure by design." And it's absolutely worth making things "secure by design" even if those features can be bypassed. The issue is not whether they can be bypassed, but whether they are there in the first place to perhaps be bypassed.

If someone goes to the effort to bypass a deliberately designed security measure, then the consequences of doing that is 100% on them. It's a matter of transferring responsibility. If something is insecure by design, then it's the designers who are at fault for making the system insecurely. You know, designing the system insecurely. They may have assumed that someone would come along and make their insecure system secure, but we've witnessed far too many instances where that never happened. So the entire world's overall net security will be increased if systems just start out being secure and are then later, in some instances, forced against their will to operate insecurely.

And if someone's manager learns that the reason the enterprise's entire network was taken over, all their crown jewels stolen and sent to a hostile foreign power, and then all their servers encrypted, is because someone in IT wedged a toothpick into a button to keep it held down for their own personal convenience.

Leo: Of course they did.

Steve: Well, you won't be asking that manager for a recommendation on the rsum that will soon need updating. Right.

Leo: Wow.

Steve: It'll be your fault and no one else's. David Sostchen tweeted: "Hi, Mr. Gibson. Longtime listener" - very formal - "and SpinRite owner." Actually, Ant used to call me Mr. Gibson, but nobody else does. He said: "I was listening to podcast 955, and I meant to message you about the Italian company Actalis, but life has a tendency to get in the way. They happen to be one of the few remaining companies that issue free S/MIME certificates. I've been using them for years to secure all my email. All the best, David." So I just wanted to pass that on, David, thank you.

Leo: That's good to know. Who is it again?

Steve: An Italian company, Actalis, A-C-T-A-L-I-S, are issuing free...

Leo: Because I've been paying for my S/MIME.

Steve: S/MIME certs.

Leo: I mean, I use PGP most of the time, which is free. But that's cool. S/MIME is a lot easier for some people, so that's cool.

Steve: Yup. Meanwhile, the FeloniousWaffle has tweeted: "Hi, Steve. I created an account on this platform to message you." Oh, thus FeloniousWaffle. He said: "I cannot wait for your email to be up and running." Neither can I. "I was just listening to Episode 968 on my commute and believe the outrage of AT&T's encryption practices to be undersold." Oh. He says: "You mention that if someone is able to decrypt one string to get the four-digit code, then they have everyone's code who shares the same string. I believe it to be far worse than that. Am I wrong in thinking that if they crack one, then they have all 10,000?

"I'm making some assumptions that there are only two ways that 10,000 unique codes produces exactly 10,000 unique encrypted strings. The first, and this is what I'm assuming, AT&T used the same key to encrypt every single code." That's right. "The second would be to have a unique key for each code. So code 123 would have to be a different key than 5678. That seems farfetched to me. Is there an error to my thinking? Thanks for the podcast and everything you do. Glad you are sticking around beyond 999. Daryle."

Okay. So I see what Daryle is thinking. He's assuming that what was done was that, if the encrypted string was decrypted to obtain the user's four-digit passcode, then the other 9,999 strings could similarly be decrypted to obtain the other four-digit passcodes. And he's probably correct in assuming that, if one string had been decrypted, then all the others could be, too. But that isn't what happened. No encrypted strings were ever decrypted, and the encryption key was never learned. But due to the static nature of the passcodes' encryption, that wasn't necessary.

I wanted to share Daryle's note because it reveals an important facet of cryptography, which is that it's not always necessary to reverse a cryptographic operation, as in decryption in this case; but it's also true of hashing, where we've talked about through the years many instances where we don't need to unhash something. You know, going only forward, in the forward direction, is often still useful. If the results of going in the forward direction can only be reapplied to other instances, then a great deal can still be learned.

In this case, since people tended to use highly non-random passcodes, you know, reusing their birthday, their house's street number, or the last four digits of their phone number or Social Security number - all things that were also part of the exfiltrated data - and assuming a fixed mapping between their plaintext passcode and its encryption, meaning the key never changed, examining, for example, the details of all of the records having a common encrypted passcode - imagine that you, from this big massive database, you pulled together all the records with the same encrypted passcode, and you look at them, just that observation would very quickly reveal what single passcode most of those otherwise unrelated records shared, and thus all of them used.

For example, one household lived at 1302 Willowbrook, whereas the birthday of someone else was February 13th, and someone else's phone number ended in 1302. So by seeing what digits were common among a large group of records all sharing only the same encrypted passcode, it would quickly become clear what identical passcode they all chose. No decryption necessary. So that's, you know, one of the cool things that we've seen about the nature of crypto in the field is there actually are some interesting ways around it when you have the right data, even if you don't have the keys.

Skynet tweeted: "Hi, Steve. Would having DRAM catch up and be fast enough eliminate the GhostRace issue?" And I thought that was a very interesting question. You know, we've talked about how caching is there to decouple slow DRAM from the processor's much more hungry need for data in a short time. So the question could be reframed a bit to further clarify what we're really asking. So let's ask: If all of the system's memory were located in the processor's most local, instant access L1 cache, that is, if its L1 cache were 16GB in size, so that no read-to or write-from main memory took any time at all, would speculative execution still present problems? And I believe the answer is yes.

Even in an environment where access to memory is not an overwhelming factor, the work of the processor itself can still be accelerated by allowing it to be more clever about how it spends its time. Today's processors, for example, are not executing instructions one at a time. And in fact processors have not actually been executing one instruction at a time for quite a while. The concept of "out of order" instruction execution dates way back to the early CDC (Control Data Corporation) 6600 mainframe, which was the first commercial computer system, a mainframe, to implement out-of-order instruction execution.

And that was in 1968, I believe, is when the CDC 6600 happened. It sucked in instructions ahead of them being needed. And when it encountered an instruction whose inputs and outputs were independent of any earlier instructions that were still being worked on, it would execute that later instruction in parallel with other ongoing work because the instruction didn't need to wait for the results of previous instructions. Nor would its effect change the results of previous instructions.

The same sort of instruction pipelining goes on today, and we would still like our processors to be faster. If a processor had perfect knowledge of the future by knowing which direction it was going to take at any branch or where a computed indirect jump was going to land it, and it could reach its - if it had perfect knowledge of those things, it would be able to reach its theoretical maximum performance given any clock rate. But since a processor's ability to predict the future is limited to what lies immediately in front of it, it must rely upon looking back at the past and using that to direct its guesses about the future - or, as we say, its speculation about its own immediate future.

Here's something to think about. The historical problem with third-party cookies has been that browsers maintained in the past a single large shared cookie jar, as we've discussed before, in fact just recently. So an advertiser could set its cookie while the user was at site "A," and read it back when the same user had moved to site "B." This was never the way cookies were meant to be used. They were meant to be used in a first-party context to allow sites to maintain state with their visitors. The problem is that, until very recently, there has been no cookie compartmentalization.

We have the same problem with microprocessor speculation that we have had with third-party cookies, lack of compartmentalization. The behavior of malware code is affected by the history of the execution of the trusted code that ran just before it. Malware is able to detect the behavior of its own code, which gives it clues into the operation of previous code that was running in the same set of processors. In other words, a lack of compartmentalization. Malicious code is sharing the same microarchitectural state as non-malicious code because today there's only one set of state. That's what needs to change. And I would be surprised if Intel wasn't already well on their way to implementing exactly this sort of change.

I have no idea how large a modern microprocessor's speculation state is today. But the only way I can see to maintain the performance we want today in an environment where our processors might be unwittingly hosting malicious code is to arrange to save and restore the microprocessor's speculation state whenever the operating system switches process contexts. It would make our systems even more complicated than they already

are, but it would mean that malicious code could no longer obtain any hints about the operation of any other code that was previously using the same system it is.

I'll omit this listener's full name since it's not important. We'll call him John. He says: "I got nailed in a phishing email for AT&T."

Leo: Oh.

Steve: Yeah, see the attached picture. He said: "No excuse, but at least I realized it immediately and changed my password," he said, "which is not one that has been used anywhere else of course." He ended, saying, "Feel stupid...."

Leo: No, because we've been talking about this AT&T breach. He was expecting this email from AT&T.

Steve: Yup. Yup. Yup. Exactly. The email says: "Dear Customer. At AT&T we prioritize the security of our customers' information and are committed to maintaining transparency in all matters related to your privacy and data protection. We are writing to inform you of a recent security incident involving a third-party vendor. Despite our rigorous security measures, unauthorized access was granted to some of our customer data stored by this vendor. This incident might have involved your names, addresses, email addresses, social security numbers, and dates of birth.

"We want to assure you that your account passwords were not exposed in this breach." But they're about to be. "We have notified federal law enforcement about the unauthorized access. Please accept our apology for this incident. To determine if your personal information was affected, we encourage you to follow the link below to log into your account."

Leo: Oh, boy.

Steve: And then there's a little highlight that says "Sign In." And finally, "Thanks for choosing us. AT&T."

Leo: I'm willing to bet this is a copy of the actual email because it's too much corporate, like, rigorous security measures, but they did gain all your data. It's very much what AT&T said. So I bet the bad guy just copied the original AT&T email and just changed this little link here.

Steve: Exactly. And, well, I would imagine that there was probably no sign-in in the original link because, you know, that's really what changes it into a phishing attack. And so anyway, I just wanted to say this is how bad it is out there. I mean, as you said, Leo, you saw it immediately. We've been talking about it. This is a listener of ours. He knew about it before it came.

Leo: He expected it. He expected it.

Steve: So absolutely authentic-looking.

Leo: They're so smart. They're so evil.

Steve: You know, we really - we absolutely need to always be vigilant.

Leo: And never click links in email.

Steve: No. No.

Leo: Never.

Steve: Even from Mom. Tom Minnick said: "With these 'atomic operations' to mitigate race conditions, how does that work with multicore processors when multiple threads are running in parallel? Couldn't a race condition still occur?" He says: "I probably don't understand enough about how multicore processors handle threads."

So Tom's question actually is a terrific one, and it occurred to many of our listeners who wrote. He and everyone were right to wonder. The atomicity of an instruction only applies to the threads running on a single core since it, the core, can only be doing one thing at a time by definition. Threads, as I said, are an abstraction for a single core. They are not an abstraction if multiple cores are sharing memory.

So what about multiple cores or multi-processor systems? The issue is important enough that all systems today provide some solution for this. In the case of the Intel architecture, there's a special instruction prefix called "Lock" which, when it immediately precedes any of the handful of instructions that might find it useful, forces the instruction that follows to also be "atomic" in the sense of multiple cores or multiple memory-sharing processors.

Only one processor at a time is able to access the targeted memory location. And after all, it's just an instant; right? It's just essentially there's actually a - there is a locked signal that comes out of the chip that all the chips are participating with. So the processor, when it's executing a locked instruction, drops that signal, performs the instruction, and immediately raises it. So it's just an, you know, it's as infinitesimally brief lockout as could be. So it doesn't hurt performance, but it prevents any other processor from accessing the same instruction at the same time. Only one processor at a time is able to access the targeted memory location.

And there's one other little tidbit. That simple exchange instruction is so universally useful for performing thread synchronization that the lock prefix functionality is built into that one instruction. All the other instructions that can be used require an explicit lock prefix. Not the exchange instruction. It automatically is not only thread safe, but multicore and multiprocessor safe, which I think is very cool.

Finally, Michael Hagberg said: "Credit Freeze. Rather than unlock your entire account, it should work this way: I'm buying a car. The dealer tells me which credit service they use and the dealership's ID number. I go to the credit service website, provide my Social Security number, PIN assigned by the site when I froze it, and the car dealer's ID number. My account will then allow that car dealer only to access my account for 24 hours."

And Michael, I agree 100%. And this just shows us that the child in you has not yet been beaten into submission, and that you are still able to dream big. More power to you. Wouldn't it be nice if the world was so well designed.

Leo: I actually do everything but that last piece where you give the car dealer's ID to the credit bureau. But I do ask them which credit bureau are you going to use.

Steve: Good, good.

Leo: And then that's the one I unfreeze. And I tell them, you've got whatever, three days to do this, and it's going to automatically lock up again. And nowadays enough people use freezes that when they get that, they kind of know what happened. And they'll call you and say, hey, your credit's frozen, yeah.

Steve: Yeah, right. It's not unusual to encounter a freeze. And in fact I did some googling around before I got my card with Amazon to find out which of the services they used.

Leo: Yeah, exactly.

Steve: And, you know, and then that's the one I unlocked so that...

Leo: I'd be more judicious, yeah. I love the idea, though, of saying, hey, credit bureau, this guy is going to ask. Don't tell anybody else.

Steve: Wouldn't that be? Yeah, but Leo, you know, all the junk mail we receive as elders...

Leo: All those credit card offers. Oh.

Steve: Yes. It's because everybody's pulling our credit.

Leo: By the way, when I froze all my accounts, I stopped getting those.

Steve: Yeah, I haven't had any for years.

Leo: The only ones I get are from cards, existing cards saying, you know, hey, you've got a blue card. Would you like a green one? That's it. Because no new card companies can get my information. So it works.

Steve: Right.

Leo: It works.

Steve: Right. Okay. We've got just two little bits regarding SpinRite. Mike Shales said: "Recently I've run into some issues with my old iMac, a mid-2017 model." He said: "I've wanted to support your valuable Security Now! efforts for some time, but investing the time to see if I could even run SpinRite on my Macs when they were all running without problems discouraged me. But now. You mentioned on your April 9th podcast: 'I wanted to remind any would-be Mac purchasers'" - oh, this is me speaking. He's quoting now. "'I wanted to remind any would-be Mac purchasers that this is the reason I created GRC's freeware named Bootable in favor of'" - okay, the name - "'In favor of 'DOS Boot.' If you can get Bootable to congratulate you on your success in booting it, then exactly the same path can be taken with SpinRite.'" Right.

So he said, he wrote: "But Bootable is a Windows .exe file and needs a Windows machine to create a bootable USB flash drive; right? Lacking a Windows machine, I made a bootable DOS drive from your ReadSpeed image download." Wow. Good going there.

Leo: It's ambitious, yeah.

Steve: "Following instructions from ChatGPT, I used dd to write the ReadSpeed image to a 4GB flash drive. Then following instructions in the GRC forum post 'Boot a Mac into FreeDOS for SpinRite or ReadSpeed,'" he said: "I succeeded in booting my iMac into DOS and running ReadSpeed. So far so good. But I believe the current SpinRite 6.1 includes the capability to recognize more drives than previously, and might rely on features not provided in the version of DOS that I now have installed on my flash drive. If so, perhaps downloading the SpinRite 6.1 .exe file and copying to my flash drive might not be ideal. Is this an issue? Thanks for the help. Mike."

Well, okay. Mike very cleverly arranged to use various tools at GRC - and, amazingly enough, ChatGPT - to create a bootable USB drive which successfully booted his mid-2017 iMac. So first responding to Mike directly: Mike, everything you did was 100% correct. And if you place your copy of the SpinRite EXE into that USB stick and boot it, everything will work perfectly. And if you run it at Level 3 on any older Macs with solid-state storage, you can expect to witness a notable and perhaps even very significant subsequent and long-lasting improvement in the system's performance.

And while it won't be obvious, there's also very a good reason to believe that in the process you'll have significantly improved the system's reliability. The reason the SSD will now be much faster is that it's needing, as I mentioned before, to struggle much less after running SpinRite to return the requested information. And we will be learning far more about this during the work on SpinRite 7. And although 6.1 is a bit of a blunt instrument in this regard, it works, and it's here today. To Mike's question, the specific version of FreeDOS does not matter at all, since DOS is only used to load SpinRite and to write its log files. Otherwise, SpinRite ignores DOS and interacts directly with the system's hardware. So yes, you can run it on your ReadSpeed drive.

I wanted to share Mike's question because I just finished making some relevant improvements. He mentioned correctly that Bootable is Windows-only freeware. But over the weekend the Bootable download was changed from an EXE to a ZIP archive, and the ZIP archive now also contains a small Bootable file system image which can be used by any Mac or Linux user to directly create a Bootable boot-testing USB drive.

Leo: Any Intel Mac or Linux user.

Steve: Yes, Intel, yes.

Leo: We should really emphasize that because most Mac users now are no longer Intel, yeah.

Steve: Yeah. Yeah. One of the guys in GRC's web forums put me onto a perfect and easy-to-use - oh, I should mention, Leo, we've solved the Intel problem, but that's for another time.

Leo: Oh. Tease me, why don't you.

Steve: Yeah. We've got - we have some guys have figured out how to boot on UEFI-only systems and on ARM-based...

Leo: On silicon, wow.

Steve: ...silicon using some concoction of virtual machines. And I haven't followed what they're doing because I'm just focused on getting what all of this has done done. Anyway, there's something known as Etcher by a company called Balena. It is a perfect, easy-to-use for an Intel Mac person means of moving the Bootable image onto a USB without the dd command. Dd makes me nervous because you need to know what you're doing. I mean, it's a very powerful direct drive copying tool. Linux people are probably more comfortable with dd. I'm glad that this Mac user, Mike, was able to get ChatGPT to help him, and I'm glad that ChatGPT just didn't stumble over hallucination at that particular moment.

Leo: You can erase everything with dd very easily.

Steve: Yeah.

Leo: Careful.

Steve: Yeah. And lastly, Sean wrote: "Hey, Steve. I'm sure you're hearing this a lot, but Windows" - oh. "Windows did not trust SpinRite despite all your signing efforts. I had to clear three severe warnings before it would allow me to keep 6.1 on my system for use. I hope it gets better soon for users less willing to ignore the scary warnings from Microsoft." Signed, Sean.

And yup. I don't recall whether I had mentioned it here also, since I've participated a lot about this over in this discussion in the GRC's newsgroups. One thing that has been learned is that Microsoft has decided to deprecate any and all special meaning for EV, extended validation, code signing certificates. It's gone. All those hoops I jumped through to get remote server-side EV code signing to work remotely on an HSM device will have no value moving forward, except having the signing key in the HSM does prevent anybody, even me, I mean, from extracting it. It can't be extracted. It can only be used to sign.

When I saw this news, I reached out to Jeremy Rowley, who's my friend and primary contact over at DigiCert, to ask him if I had read Microsoft's announcement correctly. And he confirmed that Microsoft had just, like, the week before surprised everyone in the CAB forum with that news. Apparently, what's at the crux of this is that for, you know, historically, end users were able to use EV code signing certificates to sign their kernel drivers. That was the thing Microsoft most cared about as far as EV was concerned.

But after the problems with malicious Windows drivers, Microsoft has decided to take away that right and require that only they, meaning Microsoft, will be authorized to sign Windows kernel drivers in the future. In their eyes, this eliminated the biggest reason for having and caring at all about EV code signing certs. So they will continue to be honored for code signing, but EV certs will no longer have any benefit. They will confer no extra meaning.

What I think is going on regarding SpinRite is that something Windows Defender sees inside SpinRite 6.1's code, which was not in 6.0, absolutely convinces it that this is a very specific Trojan named "Wacatac.B," which I guess you pronounce "whack attack." If I knew what part of SpinRite was triggering these false positives, I could probably change it. I have some ideas, so I'm going to see, because we just can't keep tolerating these sorts of problems from Microsoft, and it doesn't now look like my having an EV cert - it's been three months now, and tens of thousands of copies of GRC's freeware, because, you know, thousands, plural thousands of copies are being downloaded every day. I resigned them all with this new certificate in order to get it exposed and let Microsoft see that, you know, whoever was signing this wasn't producing malware. But, you know, here's Mike, or no, sorry, Sean, who just said, you know, he had to go to extreme measures to get Windows to leave this download alone.

Leo: Huh.

Steve: So, grumble.

Leo: Grumble, grumble.

Steve: Big-time. Okay. We're going to talk about what the EU is doing, Leo, after you share our last sponsor with our listeners.

Leo: Yes. Breaking news, however, that you will, depending on your point of view, will either be surprised or not surprised to hear.

Steve: Yeah.

Leo: Google has decided to delay third-party cookie blocking.

Steve: Oh.

Leo: Until next year.

Steve: Wow.

Leo: From Digiday, this fantastic opening sentence by Seb Joseph: "Google is delaying the end of third-party cookies in its Chrome browser again. In other unsurprising developments, water remains wet."

Steve: Wow.

Leo: So they did not outline a more specific timetable beyond hoping for 2025.

Steve: Okay. And that, I mean, it does show you the...

Leo: Resistance.

Steve: ...problem with taking this away.

Leo: They promised it originally in January 2020. This is the third time they've pushed it back. And I'm guessing it's not going to be the last. Some of this is actually, you know, intertwined with the UK Competition and Markets Authority. They want, they say: "It's critical the CMA has sufficient time to review all the evidence, including results from industry tests, which the CMA has asked market participants to provide by the end of June."

Steve: Ah, in order to see whether the Privacy Sandbox will be a replacement.

Leo: Right, right. "We recognize," Google says, "there are ongoing challenges related to reconciling divergent feedback from the industry. Regulators and developers will continue to engage closely with the entire ecosystem." Yeah, but some of this is that the CMA wants to see proof, and they're not ready to provide proof.

Steve: Well, Leo, here's another good - another reason I'm so happy we're going past 999.

Leo: Yeah.

Steve: Because November is when we hit 999. And I would not be here for...

Leo: Let's make a deal that you'll keep doing the show until Google phases out third-party cookies.

Steve: Oh, no.

Leo: Rats.

Steve: Well past.

Leo: Almost fooled him.

Steve: No, no. I think it's going to happen. I think it's inevitable.

Leo: You think. Okay. We'll see. It's been four years.

Steve: I would wager to 2025. I'll go for next year.

Leo: Let's go for 2025. Let's do it. Let me real quickly mention our great sponsor, and then we can get to the meat of the matter, these Chat guys here going on, what's going on in Europe. All right. Let's talk about Chat, Steve Gibson.

Steve: Okay. So, oh, boy. Across the pond from the U.S., the EU is continuing to inch forward on their controversial legislation, commonly referred to as "Chat Control," thus today's title is "Chat (out of) Control," which proposes to require providers of encrypted messaging services to somehow arrange to screen the content that's carried by those services for child sexual abuse material, commonly known as CSAM. As I said when we last looked at this last year, 2024 will prove to be quite interesting since all of this will likely be coming to a head this year. What's significant about what's going on in the EU, unlike in the UK, is that the legislation's language carries no exclusion over the feasibility of performing this scanning.

Just to remind everyone who has a day job and who might not be following these political machinations closely, last year the UK was at a similar precipice. And with their own legislation, at the 11th hour, they added some language that effectively neutered it while allowing everyone to save face. For example, last September 6th, Computer World's headline read "UK rolls back controversial encryption rules of Online Safety Bill," and followed that with "Companies will not be required to scan encrypted messages until it is 'technically feasible,' and where technology has been accredited as meeting minimum standards of accuracy in detecting only child sexual abuse and exploitation content."

So since it's unclear how any automated technology might successfully differentiate between child sexual abuse material and, for example, a photo that a concerned mother might send of her child to their doctor, there's little concern that the high bar of "technical feasibility" will be met in the foreseeable future. While the UK came under some attack for punting on this, the Big Tech companies all breathed a collective sigh of relief.

But so far - and, boy, there's not much time left - there is no sign of the same thing happening in the EU, not even a murmur of it. One of the observations we've made about all such legislation was the curious fact that, if passed, the legislation would mean that the legislator's own secure, encrypted, and private communications would similarly be subjected to surveillance and screening. Or would they?

Two weeks ago, on April 9th, the next iteration of the legislation appeared in the form of a daunting 203-page tome. Fortunately, the changes from the previous iteration were all

shown in bold type, or crossed out, or bold underlined, or crossed out and underlined, all meaning different things. But that made it at least somewhat possible to see what's changed. As you can tell, I spent way too much time with that 203 pages. This was brought to my attention by the provocative headline in an EU website, "ChatControl: EU ministers want to exempt themselves."

And what that article went on to say was: "According to the latest draft text of the controversial EU Child Sexual Abuse Regulation proposal leaked by the French news organization Contexte, which the EU member states discussed, the EU interior ministers want to exempt professional accounts of staff of intelligence agencies, police, and military from the envisioned scanning of chats and messages. The regulation should also not apply to 'confidential information' such as professional secrets. The EU governments reject the idea that the new EU Child Protection Centre should support them in the prevention of child sexual abuse and develop best practices for prevention initiatives."

Okay. So the EU has something called the "Pirate Party," which doesn't seem to be well named, but it is what it is.

Leo: No, it's a real - it's, you know, the Pirate Bay people.

Steve: Yeah, the Pirate pirates.

Leo: Yeah. It's a party of pirates, yeah. And popular, I might add.

Steve: Yes. It's formed from a collection of many member parties across and throughout the European Union. The Party was formed 10 years ago, back in 2014, with a focus upon Internet governance. So the issues created by this pending legislation is of significant interest to this group. To that end, one of the members of Parliament, Patrick Breyer, had the following to say about these recent changes to the proposed legislation which came to light when the document leaked.

He said: "The fact that the EU interior ministers want to exempt police officers, soldiers, intelligence officers, and even themselves from chat control scanning proves that they know exactly just how unreliable and dangerous the snooping algorithms are that they want to unleash on us citizens. They seem to fear that even military secrets without any link to child sexual abuse could end up in the U.S. at any time.

"The confidentiality of government communications is certainly important, but the same must apply to the protection of business and of course citizens' communications, including the spaces that victims of abuse themselves need for secure exchanges and therapy. We know that most of the chats leaked by today's voluntary snooping algorithms are of no relevance to the police, for example family photos or consensual sexting. It is outrageous that the EU interior ministers themselves do not want to suffer the consequences of the destruction of digital privacy of correspondence and secure encryption that they are imposing upon us.

"The promise that professional secrets should not be affected by chat control is a lie cast in paragraphs. No provider and no algorithm can know or determine whether a chat is being conducted by doctors, therapists, lawyers, defense lawyers, et cetera, so as to exempt it from chat control. Chat control inevitably threatens to leak intimate photos sent for medical purposes and trial documents sent for defending abuse victims. It makes a mockery of the official goal of child protection that the EU interior ministers reject the development of best practices for preventing child sexual abuse.

"It couldn't be clearer that the aim of this bill is China-style mass surveillance, and not better protecting our children. Real child protection would require a systematic evaluation and implementation of multidisciplinary prevention programs, as well as Europe-wide standards and guidelines for criminal investigations into child abuse, including the identification of victims and the necessary technical means. None of this is planned by the EU interior ministers."

So after the article finished quoting Patrick Breyer, it noted that the EU governments want to adopt the chat control bill by the beginning of June. We're approaching the end of April, so the only thing separating us from June is the month of May. I was curious to see whether the breadth of the exclusion might have been overstated in order to make a point, so I found the newly added section of the legislation on page 6 of the 203-page PDF. It reads - this is Section 12a. The "a" is the new part.

"In the light of the more limited risk of their use for the purpose of child sexual abuse and the need to preserve confidential information, including classified information, information covered by professional secrecy and trade secrets, electronic communications services that are not publicly available" - that's the key. "Electronic communications services that are not publicly available, such as those used for national security purposes, should be excluded from the scope of this Regulation. Accordingly, this Regulation should not apply to interpersonal communications services that are not available to the general public, and the use of which is instead restricted to persons involved in the activities of a particular company, organization, body, or authority."

Okay, now, I'm not trained in the law, but that doesn't sound to me like an exclusion for legislators who would probably be using iMessage, Messenger, Signal, Telegram, WhatsApp, et cetera. It says, "This Regulation should not apply to interpersonal communications services that are not available to the general public." So, you know, internal proprietary intelligence agency communication software, you know, applications.

Remember that it's this proposed EU legislation which includes the detection of "grooming" behavior in textual content. So it's not just imagery that needs to be scanned, but the content of all text messaging. We're also not talking about only previously known and identified content which is apparently circulating online, but also anything the legislation considers "new" content. As I read through section after section of what has become a huge mess of extremely weak language that leaves itself open to whatever interpretation anyone might want to give, my own lay feeling is that this promises to create a huge mess. I've included a link to the latest legislation's PDF in the last page of the show notes for anyone who's interested.

You'll only need to read the first eight pages or so to get a sense for just what a catastrophic mess this promises to be. As is the case with all such legislation, what the lawmakers say they want, and via this legislation will finally be requiring, is not technically possible. They want detection of previously unknown imagery and textual dialog which might be seducing children while at the same time honoring and actively enforcing EU citizen privacy rights. Oh, and did I mention that 78% of the EU population that was polled said they did not want any of this?

And it occurred to me that encryption providers will not just be able to say they're complying when they are not, because activist children's rights groups will be able to trivially test any and all private communications services to verify that they do, in fact, detect and take the action that the legislation requires of them. All that's needed is for such groups to register a device as being used by a child, then proceed to have a pair of adults hold a seductive grooming conversation and perhaps escalate that to sending some naughty photos back and forth. And you can believe that, if the service they're testing doesn't quickly identify and red-flag the communicating parties involved, those

activist children's rights groups will be going public with the service's failure under this new legislation.

I've said it before, and I understand that it can sound like an excuse and a copout, but not all problems have good solutions. There are problems that are fundamentally intractable. This entire debate surrounding the abuse of the absolute privacy created by modern encryption is one such problem. This is not technology's fault. Technology simply makes much greater levels of privacy practical, and people continually indicate that's what they prefer. As a society we have to decide whether we want the true privacy that encryption offers, or whether we want to deliberately water it down in order to perhaps prevent some of the abuse that absolute privacy also protects.

Leo: Agreed, agreed, and agreed.

Steve: Yeah. I do commend to anyone, the last page of the show notes has a link. It's not widely available publicly because it was leaked, and Patrick-Breyer.de, so a German site, has it on his site and is making it available. So you'll need to get it if you're interested. But, boy, as I said, just reading through it, it is, again, it's insanely long at 203 pages. I struggled to find any language about, like, what time period this takes effect over. I couldn't find any. It all seems to indicate, once this legislation is in place, that the organizations need to act.

But I just think the EU is stepping into a huge mess. And again, as I said, 2024, I said last year this next year, 2024 we're in now, is going to be one to watch because lots of this is beginning to come to a head. Although, Leo, as you just shared with us, not the third-party cookie issue with Chrome. That's been punted into when we have four digits on this podcast.

Leo: In the future. Ah, yeah. It's an interesting world we live in.

Steve: So I imagine when the legislation happens, and it's supposed to be happening in early June, there will be lots of coverage. We'll be back to it, and we'll know, you know, we'll have some sense for when it's taking effect and what the various companies are choosing to do.

Leo: Yeah, they might be well modified from this leaked document. There will certainly be amendments and things like that. So we'll have to look at the actual legislation to see what's happening.

Steve: Right.

Leo: And we will because that's what we do. That's what we do here.