Secure Conversation Records Retention

Description: The state of Virginia passes an age-restriction law that has no chance. New Zealand also tries something similar, citing Australia's lead. A nasty Python package for Discord survived three years and 11,000 downloads. The FBI says it's a good idea to discard end-of-life consumer routers. What's in WhatsApp? Finding out was neither easy nor certain. The UK's Cyber Centre says AI promises to make things much worse. A bunch of great feedback from our great listeners. Then, is true end-to-end encryption possible when records must be retained?

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-1025.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-1025-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We've got a lot of security news. The state of Virginia passes an age-restriction law that Steve says has no chance of surviving a First Amendment challenge. There is a nasty PyPI package that has survived for three years and 11,000 downloads. Isn't anybody paying attention? And then Steve has a solution for government agencies that want to use Signal without letting the whole world know what they're talking about. All of that and more coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 1025, recorded Tuesday, May 13th, 2025: Secure Conversation Records Retention.

It's time for Security Now!, the show where we cover your security, your privacy, your safety online with the guy in charge at GRC.com, Mr. Steve Gibson. Hello, Steve.

Steve Gibson: I have never been accused of talking quietly, Leo. Never been anything that has happened. My wife has extremely sensitive ears, and she talks from the other room and assumes I can hear her. She can hear me. So it's not reciprocal. But, you know.

Leo: This is a, you know, this is generally a couple problem. Lisa will talk to me from two floors away.

Steve: I'm glad it's not just me. And I can hear her. It's like, what? It's like something's being - there's a conversation being had, and she doesn't talk to herself, so I know it's aimed. But...

Leo: I wear hearing aids, and it still doesn't help. So I don't know what the answer is.

Steve: Oh, but you have such an excuse. That's great. You know, it's like, they were off, or they...

Leo: Eh? I need a trumpet. I need one of those old-timey - eh?

Steve: Anyway.

Leo: So what's coming up this week on Security Now!?

Steve: We are here for episode, the big episode, first episode past the 2^10, Episode 1025 for the 13th of May. I got to today's topic through a rather circuitous route because it was originally titled "What Is End-to-End Encryption?" Because I was thinking, I was noticing that it's become a buzzword. It's become, like, oh, it's what you're supposed to have, or oh, it's this. Oh, don't worry about it, it's end-to-end encrypted. Even when, you know, like it may not be.

And of course this harkens back to the TeleMessage mess that we talked about, and them insecurely archiving Signal chat messages, Signal protocol conversations. But as I looked further, and I thought about this problem, I ended up sort of - the podcast morphed into today's topic, which is "Secure Conversation Records Retention." Because that's really sort of the issue and the question. And the question is, is true end-to-end encryption possible when records are being retained?

Leo: Yeah, because records have to be kept in the clear; right? Or at least - I guess not.

Steve: Well, records have to be accessible if you are subpoenaed for...

Leo: Right, right.

Steve: And, like, I remember being appalled when my little - when I heard, like, my little company that's sort of doing nothing big over in the corner, you know, might have to produce email records if someone were ever to sue us for something. It's like, what? It's our email. No, it's, you know, it's corporate records responsibility. And there's all kinds of, like, record-keeping acts now. And in fact since I sent this email, the email of this show with the show notes to, I think now we're up to 17,363 of our listeners, several people wrote back and said that they were, their enterprises, their companies were TeleMessage customers, and that they're now needing to find an alternative solution because this actually is a problem. There is a need that companies have for their executives' dialogues and conversations and transactions to be retained for legal purposes.

Leo: Sure. Yeah. There's regulations in almost all industries about that.

Steve: Anyway, so we're going to get around to that and I think have an interesting exploration of the issue and the problem. And believe it or not, Leo, I have a solution.

Leo: Oh.

Steve: Which I didn't have it when I started. It was as a consequence of sort of brainstorming during the podcast production and putting it all down. And I hope it happens. I imagine it will because, you know, the podcast has some reach. And if it hasn't occurred, maybe it's occurred to other people, but I have not seen it anywhere.

Leo: Oh, I love this.

Steve: So we'll talk about that.

Leo: Oh, that's great.

Steve: First we'll talk about the state of Virginia passing an age-restriction law that, as I wrote, has no chance. It's like, what are you - why are you even bothering to waste the ink on this, you idiots? Also, New Zealand also tries something similar, citing the lead that they're taking from Australia. We have a nasty Python package that actually is aimed at Discord developers which was only found after three years and more than 11,000 downloads. Which tells us the story of, well, we can't count on all the security firms always finding all of the malicious, you know, repository junk that's out there. Also, what's in WhatsApp? Turns out that finding out was neither easy nor certain. And there's a story there. Also the UK's Cyber Centre says that AI promises to make things much worse. Oh, joy.

We've got a bunch of great feedback from our listeners which we're going to spend some time on, and also use those as talking points. And then look at this question of what does it mean to need to retain records of what originally was a secure end-to-end encrypted conversation, and how could that be done? And of course the Picture of the Week also generated feedback from our listeners. Benito and I were talking about this beforehand as MacBreak Weekly was wrapping up. I learned something I didn't know which is interesting from our listeners and which Benito also told me. But the picture is interesting. So we'll have fun looking at it.

Leo: All right. I haven't looked at it, as always. I like to save it for the show. All right. I'm ready to scroll up. I have been very good, you know, it's hard for me. I had it right in front of me.

Steve: I know. We appreciate having your candid first look, as they say. So I gave this picture the caption "A joke? Serious? Deliberately setting a high bar? Or maybe missing the point of deliberately posting a WiFi password?"

Leo: That is the most annoying password I've ever seen.

Steve: Okay.

Leo: It's secure.

Steve: So for our listeners who are not seeing this, first we have a guy who enjoyed, or a person, I don't know who, you know, what their sex was, a person who had a lot of fun with fonts. This is a framed, looks like 8.5x11 gray sheet. And actually it's a true photo. You can barely see the reflection...

Leo: Oh, yeah, yeah, yeah.

Steve: ...of the person's smartphone that bounced off the glass who was taking a picture of this because they're like, what the heck? Anyway, this says "Welcome" in a nice big scroll-y font. Then it says "Be Our Guest" in a sans-serif ital font. Then they went to great trouble to find the very familiar WiFi icon, you know, the dot with the three radiating semicircles coming off of it, so you know that this is about the WiFi. And so they're saying, by all means, connect. Now, then we hit the problem here. The network ID, for a reason that's not clear, is TIM, T-I-M, in all caps, hyphen 92494870. Instead of being, you know...

Leo: Does that mean, wait a minute, that must mean there are 92 million other TIMs. Is that what we're...

Steve: You know, it could just be Barney or something. I mean...

Leo: It doesn't matter.

Steve: What is 92494870 following TIM?

Leo: Any normal...

Steve: But that's not the - that's not the worst, Leo.

Leo: No? It's worse? It's worse.

Steve: Because the password is 9FzHAfcAtcZ5Rb6RuSE3YE3G.

Leo: Nicely done. Very nice.

Steve: And, okay. And so here's of course my point. It's like, what? Be our guest, you know, first of all, there's no chance that you could type this into any kind of a touch screen, you know, in your phone now. The email that I received, as I mentioned, after people saw this, they said, well, you know, Steve, the nice thing about today's smartphones is you can just aim the camera at that.

Leo: Oh, that's true.

Steve: Take a picture of it.

Leo: But, you know, you could even - I'll do you one better. I have a QR code on my wall so visitors can - and they just scan the QR code, and it joins the WiFi. You don't have to type anything in at all.

Steve: Again. So, okay. Now, so here's my question.

Leo: So this is a joke. Especially because at the bottom it says please enter the characters of the letters as they are written. It's a joke.

Steve: Maybe they're just very OCD? I don't know.

Leo: He's teasing people, I think.

Steve: And so that's my question, exactly. So maybe, and so, you know, welcome, be our guest...

Leo: Type this in.

Steve: Here is your typing test.

Leo: Yeah, that's crazy.

Steve: Anyway, I did enjoy the picture. And, you know, looking at this, I don't have it in front of me, but I found the most perfect perspective correction app for Windows. The photo that was sent to me was taken way off axis in both directions. So it was a skewed trapezoid from hell. And I found this app some time ago which works exactly the way you would want it to. But nobody else seems to have figured this out. It lets you, it gives you a four-point rubber-banded rectangle where you simply, you know, drag the four corners of the rectangle to four corners which should be rectangular on the image, and then it fixes it. And so what we're looking at in this perfectly, looks like it's exactly square-on image, originally really skewed.

Anyway, I'll get the name of it for next week because it's just the best thing. And it seems to be, you know, Apple has all this weird, like, you know, you swing it up and down and back and forth and try to negotiate with it. But when you know that what you want is something square, just drag a square over it. Anyway, I don't know why nobody else has done that. I've never found it on a Windows app. And of course now 25 of our listeners will say, "Steve, here's the one that I use." It's like, okay, thank you.

Okay. So my title for this first piece of news was "Virginia's Folly," and I wasn't sure whether to file this under "Things that will never happen" or "Good luck with that" because the event was so obviously fraught. And there was a ton of coverage about it

because, I mean, like when I did a little bit of googling, it was widely covered, probably because everyone recognizes this is really an issue. And we've talked about this a lot. But I found a very clear and concise blog posting among all the other newsy stuff from a law firm, Hunton Andrews Kurth, who specialize in privacy and cybersecurity law. So they're watching these sorts of things happen from their perspective as a group of attorneys. The headline title of their posting was "Virginia Governor Signs into Law Bill Restricting Minors' Use of Social Media."

And then they explain: "On May 2nd, 2025, Virginia Governor Glenn Youngkin signed into law a bill that amends the Virginia Consumer Data Protection Act (VCDPA) to impose significant restrictions on minors' use of social media. The bill comes on the heels of recent children's privacy amendments to the VCDPA that took effect on January 1st, 2025." So beginning of this year. They wrote: "The bill amends the VCDPA to require social media platform operators to, first, use commercially reasonable methods such as a neutral age screen to determine whether a user is a minor under the age of 16; and, second, limit a minor's use of the social media platform to one hour per day, unless a parent consents to increase the daily limit.

"The bill prohibits social media platform operators from using the information collected to determine a user's age for any other purpose. Notably, the bill also requires controllers and processors to treat a user as a minor under 16 if the user's device 'communicates or signals that the user is or shall be treated as a minor,' including through 'a browser plug-in or privacy setting, device setting, or other mechanism.' The bill also prohibits social media platforms from altering the quality or price of any social media service due to the law's time use restrictions.

"The bill defines 'social media platform' as a 'public or semipublic Internet-based service or application' with users in Virginia that connects users in order to allow users to interact socially with each other within such service or application; and allows users to do all of the following. And there's three: construct a public or semipublic profile for purposes of signing into and using such service or application; populate a public list of other users with whom such user shares a social connection within such service or application; and, finally, create or post content viewable by other users, including content on message boards, in chat rooms, or through a landing page or main feed that presents the user with content generated by other users.

"The bill exempts from the definition of 'social media platform' a service or application that, first, exclusively provides email or direct messaging services; or, two, consists primarily of news, sports, entertainment, ecommerce, or content preselected by the provider and not generated by users, and for which any chat, comments, or interactive functionality is incidental to, directly related to, or dependent on the provision of such content.

"The Virginia legislature declined to adopt recommendations by the Governor that would have strengthened the bill's children's privacy protections. These amendments to the VCDPA take effect on January 1st of next year, 2026." So the last changes to the VCDPA took effect on January 1st this year. This is setting things up for the beginning of next year.

Now, this legislation won't get off the ground before it is enjoined by multiple lawsuits arguing, with some strong rationale and probably merit, that the imposition of these restrictions flies directly in the face of the freedom of speech rights enshrined by the First Amendment to the U.S. Constitution. I mean, these things are always immediately sued, and then they go into the courts.

And it was for that reason that I recently mentioned I was hoping that the nine justices on our Supreme Court enjoy working, since what we're seeing is the upper echelons of

the U.S. legal system are being put to much more use than they've seen in quite a while. You know, how many times have we recently heard "This will eventually need to be decided by the Supreme Court?"

Leo: That's true, yeah, yeah.

Steve: You know? We're no longer talking about whether party A defrauded party B. Those are easy things to decide, relatively. Now we're asking where, exactly, the line can be drawn when a state wishes to restrict what can reasonably be described as the free speech rights of a group of individuals. Consequently, many fundamental questions surrounding proposed laws and their precise relationship to the U.S. Constitution are now being asked, and they will eventually be tried. So wow. I mean, the system is being stressed; but we need to see, you know, what the answers are that come out the other end.

So anyway, one more point on this. In New Zealand, the New Zealand press writes: "The National Party wants to ban 16 year olds" - now, I think that Virginia said "under 16." So now New Zealand has 16 and under. So again, this is why in my opinion Apple trying to create zones of age, I don't know why, to maybe make it less obvious when someone's birthday is? You know, okay. But the problem is there's no alignment of any of these laws, not even among states, let alone among nations. So, you know, I just think giving the phone an API that lets the app say, you know, is the person above or below a given age is what we're going to end up with.

Anyway: "The National Party wants to ban 16 year olds from accessing social media" - apparently all - "by forcing companies to use age verification measures." So this is, you know, the country of New Zealand. "National MP Catherine Wedd, with the backing of leader Christopher Luxon, has put forward a members' bill which would follow Australia's lead on cracking down on social media giants.

"The Prime Minister said he wanted to explore picking it up as a 'broader government bill'" - which is actually a term that means something within their legal framework - "'which would mean it could become law more quickly. Right now the legislation does not have government endorsement, which means it would be debated only if it was drawn from the ballot at random." Which seems bizarre, but okay.

"Catherine Wedd said the Bill would put the onus on social media companies to verify someone is over the age of 16 before they access social media platforms, and is modeled off Australian legislation." Wedd said: "Currently, there are no legally enforceable age verification measures for social media platforms in New Zealand." She said she'd heard from parents, teachers and principals that there wasn't enough protection in place.

"My Social Media Age-Appropriate Users Bill is about protecting young people from bullying, inappropriate content, and social media addiction by restricting access for under 16 year olds." Now she says "under 16 year olds." So okay. But before it was over the age of 16. So even they're not sure. "The bill would require social media platforms to take 'all reasonable steps' to prevent under-16s from creating accounts. It would also introduce 'penalties for non-compliance,' including financial ones."

So here's another piece. You know, this bit of news now from New Zealand reminds us that Australia has recently been exploring similar legislation. So stepping back from this, I would say that it should be very clear to anyone, you know, who's watching everything that's going on, that sooner or later, and apparently sooner, we're going to be seeing age-based restrictions on access to social media which have until now been completely uncontrolled. You know, and we know that the likes of Meta don't want it to be their

responsibility. They're saying it should be the vendor, the platform producer's responsibility. So it's a mess. And, you know, we're seeing legislation being put in place. It's being fought back against by those who don't want it to be done that way. We have seen Apple make some steps forward. We've seen Google make some steps forward. So we need...

Leo: I think you made an excellent point, though. Apple could solve this right away just by putting that feature in that you suggested, which is the parent says this kid is such and such age.

Steve: Right.

Leo: And that would be the best way to do it. Make it the parents' choice.

Steve: Right.

Leo: And Apple just needs to give - and Google need to give them that facility, that capability.

Steve: Yeah. All you have to do is surface an API. And that would mean that the Worldwide Web Consortium would define a new JavaScript verb which allowed the browser to query the platform for its user's age.

Leo: Right. You also, legislators also have to realize you can't keep kids away from all hazardous content.

Steve: Well, Leo.

Leo: You just can't.

Steve: That's the other - that's the other point is, like, nice to have a law.

Leo: Right.

Steve: But, you know?

Leo: There's friends, neighbors. Hey, I used to go down to the local drug store and read the Playboys on the shelf. I mean, there's no way you can keep them safe with a law.

Steve: Yeah. Yeah. I'm continually seeing reports of malicious supply chain attacks where this security company discovers 52 malicious Python libraries or that security company finds 46 malicious JavaScript libraries. And, I mean, I don't talk about them

every week because I see them every week. And it's like, okay, okay. I mean, so I feel like I share them enough with our listeners to keep everybody, you know, aware of the issue, to keep it in mind, to always bring some caution to going into a repository and saying, oh, look, this is just the thing, this is just the library that I was looking for. I'm so glad someone created this.

Because, you know, the bad guys have figured out that that's what people do. And they've figured out that they can use the inherent openness of this ecosystem that we've created to hurt people, to get into people's computers, to infect somebody who's in an enterprise, then pivot from their PC when it's connected to the enterprise network into the enterprise's network, and before you know it they're listed in that ransomware listing site, and there's trouble.

So again, I guess I want to make it clear that it's not like that only happens when I mention it. It's very much like the ransomware attacks. I see 12 of them every single week. This company, that company, it's like, okay, well, that's just boring now because it's in the background. It's constant noise happening. So is all of the repository infection and the security companies, bless their hearts, that are taking their time, you know, it must just be PR for them; right? They're like, well, we found 52 malicious Python libraries. Woohoo! You know? Pay attention to us. Oh, but our customers weren't affected because our scanners, you know, nipped those in the bud. So it's a way of saying, and if you were one of our customers, you know, you'd have our scanner in your PC, too; and those 52 nasties would have never had a chance to get going.

But the troubling question is do they find them all? And what would not finding some look like? And that's the reason that Socket Research's reporting of a very malicious and sophisticated Python Trojan which had remained unfound, hidden for more than three years, was troubling. Their research posting was titled "Malicious PyPI Package Targets Discord Developers with a Remote Access Trojan," and their subtitle was "The Socket Research team investigates a malicious Python package disguised as a Discord error logger that executes remote commands and exfiltrates data via a covert command-and-control channel."

So I'll just share the start of their long report. They wrote: "On March 21st, 2022" - thus the more than three years ago part. "On March 21st, 2022 a Python package 'discordpydebug'" - all one word, discordpydebug - "was uploaded to the Python Package Index (PyPI) under the name 'Discord py error logger.' At first glance, it appeared to be a simple utility aimed at developers working on Discord bots using the Discord.py library. However, the package concealed a fully functional remote access trojan (RAT). Over time, the package received over 11,000 downloads, placing thousands of developer systems at risk.

"The package targeted developers who build or maintain Discord bots, typically indie developers, automation engineers, or small teams who might install such tools without extensive scrutiny. Since PyPI doesn't enforce deep security audits of uploaded packages, attackers often take advantage of this by using misleading descriptions, legitimate-sounding names, or even copying code from popular projects to appear trustworthy. In this case, the goal was to lure unsuspecting developers into installing a backdoor disguised as a debugging aid.

"Discord's developer ecosystem is both massive and tightly knit. With over 200 million monthly active users, more than 25% of whom interact with third-party apps, Discord has rapidly evolved into a platform where developers not only build, but also live test, share, and iterate on new ideas directly with their users. Public and private servers dedicated to development topics foster an informal, highly social culture where tips, tools, and code snippets are shared freely and often used with little scrutiny. It's within these trusted peer-to-peer spaces that threat actors can exploit social engineering

tactics, positioning themselves as helpful community members and promoting tools like discordpydebug under the guise of debugging utilities that they're familiar with.

"The fact that this package was downloaded over 11,000 times, despite having no README or documentation, highlights how quickly trust can be weaponized in these environments. Whether spread via casual recommendation, targeted DMs, or Discord server threads, such packages can gain traction before ever being formally vetted."

So the link to the rest of their extensive research is in the show notes for anyone who's interested. They talk about it at great length. But my intent here was to just, you know, without wanting to shut down the value that can be obtained, to really put a point on the fact that, you know, as they say, there's no such thing as a free lunch. I mean, this ecosystem that has been created is so neat, I mean, it's so cool to be able to have access to other people's freely shared work like this. But unfortunately, they're other people we don't know, and they can be other people who are trying to hurt us, trying to, you know, again, typically not us as individuals. But we're the - now, you know, Discord developers may well be working with an enterprise and using Discord in an enterprise environment. The bad guys want to get in there.

And so, you know, end users are no longer the targets that we once were. As soon as cryptocurrency happened and the concept of ransomware happened, and encrypting servers and exfiltrating proprietary data and threatening in return for money happened, suddenly all the attention went to how do we get in? And that's through phishing, and that's through contaminating repositories and using those as the backdoors into more valuable networks. So, you know, it is really happening. And this thing sat there for three years without being found.

Leo: Wow. Do they know how many people downloaded it, installed it?

Steve: More than 11,000.

Leo: Wow. That's terrible.

Steve: Yeah.

Leo: It's funny because they weren't, I mean, I guess it was indirectly targeting Discord because they were going after the developers; right? But I guess once you compromise a developer you can then compromise their code.

Steve: Well, yes, yes. It's the developer, the profile of a developer is one who is looking in the Python libraries for things that will help their work. And so, you know, a Discord debugger would be something that a Discord bot developer would say, hey, that's great, I want help debugging my Discord app. And so it's on a developer machine who then, you know, works for an enterprise, or maybe he's doing that on his own in the evening on his laptop, and then brings it into the office, plugs it into the network, and now this thing is able to move laterally into their employer's network, which is what they really want.

And you know, Leo, what our listeners really want...

Leo: No, they don't. But I'm going to do it anyway. What is that little thing you've got there? That is cool. That's like an eInk...

Steve: I found out - it was on some podcast of TWiT's. And it's called Terminal.

Leo: Oh, yeah.

Steve: One of your hosts mentioned it.

Leo: Yes.

Steve: CRMN...

Leo: You have your Google Analytics for your website on there?

Steve: Yeah.

Leo: That's so cool.

Steve: Yeah. And you can see on April, what is that, 28th, I think, somebody mentioned ValiDrive. And so we had a traffic...

Leo: Oh, big spike, yeah.

Steve: ...spike then. But it is, it's battery powered. It runs for three months because it uses eInk, and it updates occasionally. They have a whole bunch of little widgets that you're about to drag so you...

Leo: Super cool.

Steve: ...can compose the screen that you want. And it sits running on batteries, as I said, about three or four months, and it updates occasionally. And there's also, the reason I was interested, is that they have a developer kit that allows you to basically have it display a web page. And so at some point I'm going to have it monitoring, you know, internal server stuff.

Leo: Well, as long as we're celebrating, I might as well show you that you were a TikTok star last week.

Steve: What?

Leo: You talked about Microsoft's bold Passkey move.

Steve: Ah.

Leo: You know, getting rid of passwords. And it must have made it to the For You page on TikTok because you see your 193,000 views, 2,864 likes, 322 comments. That is very - those are good numbers. For TikTok that's incredible.

Steve: Interesting.

Leo: Yeah. So congratulations.

Steve: Thank you for reposting that.

Leo: You can now say, "Ah, yeah, I'm a TikTok star," along with everything else you tell your neighbors. Yeah, that's - yeah, that's me.

Steve: Yeah, I don't tell them that. They'd, like, look at me and go, what?

Leo: They go, "You're a what?"

Steve: Our neighbors have no idea with either Lorrie or I do. They just know we're nice people.

Leo: Yes, that's good.

Steve: But, you know, Lorrie starts talking about neurofeedback and modifying brainwaves, and they're like, what?

Leo: Well, that's all the rage nowadays, though. That's a hot topic. All right. My mouse is stuck. Oh, there we go. I finally got rid of the screen. Go ahead, my friend, go ahead.

Steve: Okay. So once upon a time it may have been difficult to toss a perfectly good consumer router into the trash bin. And while it's still probably not easy or reflective, last Wednesday the U.S. Federal Bureau of Investigation, our FBI, posted one of their PSAs, a Public Service Announcement which was titled "Cyber Criminal Proxy Services Exploiting End of Life Routers."

Here's what the FBI wrote. They said: "The Federal Bureau of Investigation (FBI) is issuing this announcement to inform individuals and businesses about proxy services taking advantage of end-of-life routers that are susceptible to vulnerabilities." And then they explain: "When a hardware device is end of life, the manufacturer no longer sells the product and is not actively supporting the hardware, which also means they are no

longer releasing software updates or security patches for the device. Routers dated 2010 or earlier likely no longer receive software updates issued by the manufacturer and could be compromised by cyber actors exploiting known vulnerabilities.

"End-of-life routers were breached by cyber actors using variants of TheMoon malware botnet. Recently, some routers at end of life, with remote administration turned on, were identified as compromised by a new variant of TheMoon malware. This malware allows cyber actors to install proxies on unsuspecting victim routers and conduct cybercrimes anonymously.

"A proxy server is a system or router that provides a gateway between users and the Internet. It is an intermediary between end-users and the web pages they visit online. A proxy is a service that relays users' Internet traffic while hiding the link between users and their activity. Cyber actors use proxy services to hide their identities and location. When actors use a proxy service to visit a website to conduct criminal activity, like stealing cryptocurrency or contracting illegal services, the website does not register their real IP address and instead registers the proxy IP.

"TheMoon malware was first discovered on compromised routers in 2014 and has since gone through several campaigns. TheMoon does not require a password to infect routers; it scans for open ports and sends a command to a vulnerable script. The malware contacts the command-and-control server, and the C2 server responds with instructions, which may include instructing the infected machine to scan for other vulnerable routers to spread the infection and expand the network.

"Tips to Protect Yourself," they wrote. "Commonly identified signs of malware infections on routers include overheated devices" - yeah, like when it's mining cryptocurrency with abandon - "problems with connectivity [same], and changes to settings the administrator does not recognize. The FBI recommends individuals and companies take the following precautions." And they list five.

"If the router is at end of life, replace the device with an updated model, if possible. Second, immediately apply any available security patches and/or firmware updates for your devices. Third, login online to the router settings and disable remote management/remote administration, save the change, and reboot the router. Fourth, use strong passwords that are unique and random and contain at least 16, but no more than 64 characters. Avoid reusing passwords and disable password hints. And finally, if you believe there is suspicious activity on any device, apply any necessary security and firmware updates, change your password, and reboot the router."

So, you know, this is good but not surprising advice for anyone listening to this podcast. Still, it's not anything that most non-cybersecurity aware users would ever think to consider. So it's a good thing that these sorts of reminders and advisory Public Service Announcements are being made by an entity that the public would trust, like the FBI. So that's good.

Three UK researchers, two from King's College and the third from Royal Holloway University of London, decided to tear WhatsApp apart to figure out how it solves the challenges of multi-device group messaging, and to see whether they may have left any rough edges in there. Here's how they described their work in their resulting paper's Abstract, which was short. The Abstract was short; the paper was not.

They wrote: "WhatsApp provides end-to-end encrypted messaging to over two billion users. However, due to a lack of public documentation and source code, the specific security guarantees it provides are unclear. Seeking to rectify this situation, we combine the limited public documentation with information we gather through reverse-engineering its implementation to provide a formal description of the subset of WhatsApp that

provides multi-device group messaging. We utilize this description to state and prove the security guarantees that this subset of WhatsApp provides. Our analysis is performed within a variant of the Device-Oriented Group Messaging model, which we extend to support device revocation. We discuss how to interpret these results, including the security WhatsApp provides, as well as its limitations."

Okay. Now, that was their Abstract. What followed was a quite daunting 115-page paper. And remember, you know, we typically encounter papers like this that are 16 to 29 or 30 pages. This is a monster. And they start off by explaining: "Group messaging in WhatsApp is based on the Signal two-party protocol and the Sender Keys multiparty extension. To date, in the academic literature, the ground truth for answering the question of how these building blocks are composed precisely is established by the WhatsApp security whitepaper or unofficial third-party protocol implementations."

Now, I'm not going to go into 115 pages because, I mean, this is really hair-curling stuff. But it put me in mind of how spoiled I guess I am. I know, Leo, you are, and many of our listeners are by Unix and Linux and Signal and other open source security systems and operating systems. The idea that these researchers were forced to reverse engineer and divine the operating protocol of a critical encrypted communications application such as WhatsApp, which is in use, as they said, by more than two billion people, seems really wrong. You know?

Now, it's true that my own SQRL client for Windows was not open source, but every single detail about the protocols it implemented and how it did them was scrupulously detailed for the specific purpose of facilitating independent implementations, and all I was doing was creating an implementation of SQRL's specification that was in the public domain from the first moment I disclosed its operation here on the podcast. And we know that the specification was correct because a number of other people created their own fully working SQRL implementations and everything interoperated perfectly. That's the way these sorts of things should be done.

Now, the details of the security protocols that, in the case of WhatsApp, as I said, billions of people depend upon, that should not be considered proprietary. That's old thinking. It's like outlawing the export of ciphers using more than 40 bits. It's not the crypto way. While I was following the references in this paper, I got a kick out of noticing that they were sorted alphabetically, which brought all of the people's names together. And at the M's we have Reference 50.

Leo: Moxie?

Steve: Marlinspike, M.: Private Group Messaging, dated May 2014, and a link.

Leo: That's the Signal protocol.

Steve: Yup. Reference 51, Marlinspike, M.: The Double Ratchet Algorithm, November 2016, and a link to the full specification revision 1. Reference 52, Marlinspike, M.: The X3DH Key Agreement Protocol, November 2016, and a link to the full specification revision 1. And Reference 53, Marlinspike, M.: The Sesame Algorithm: Session Management for Asynchronous Message Encryption, April of 2017, and a link to the full specification reference revision 2. Our listeners who've been with us from the early days may recognize every one of those specifications and protocols. And I heard you recognizing them, Leo.

Leo: Oh, yeah.

Steve: Because we have examined each of them over the years as we've followed Moxie Marlinspike's work from the start. And this is the point I wanted to make. The fact that Moxie and Signal have been sharing all of the details of their work all along demonstrates a fully mature understanding of security. That's the only way they can know what they have done is secure is by publishing it for other academic researchers to examine. By comparison, the fact that Meta's WhatsApp has taken advantage of all the good parts of that work for free, while refusing to disclose the very important workings of their own "proprietary" - and I have that in quotes - extensions of that work, is what I have a difficult time excusing.

After slogging through the 115 pages of the researchers' work - and remember, that's their output; they had to do the work in order to produce 115 pages of detailed results - they summarize their findings mostly by explaining that they were unable to find anything big that seemed to be amiss. But since they didn't have access to any source code or even to complete algorithm descriptions, "We couldn't find anything" is not the same as "We looked at everything that matters, and it all looks fine." You know, there were a number of edge cases that they were unable to explore due to lack of information, and some others that they found that were not hugely concerning. But, yeah, okay, like revoking the keys from a device when it's coming out of a group chat is a problem. And not exactly clear that WhatsApp has done that in a very solid fashion.

So these intrepid academics, you know, did achieve something for their efforts, though what is so annoying is their task could and should have been so much easier. And nothing is gained by Meta hiding what they've done and considering it, oh, these are, you know, secret sauce. No. It's two billion people are depending upon you not having made a mistake. And what we know is mistakes happen. And, you know, they need to be aired in order to get fixed.

Oh, Leo, you're going to love this. The UK's big National Cyber Security Centre (NCSC), which is roughly their equivalent of our CISA, just issued a report looking at the probable effect AI is expected to have upon cybersecurity over the next two years - from now until 2027. Now, I have to say I was somewhat relieved to see that that was their timeframe, to see that they clipped this to a relatively short-term window, since AI, arguably, is advancing so rapidly that any attempt to say anything meaningful about, for example, the next 10 years would be little more than a flight of fancy. No one has any remote clue what the AI of 2037 is going to look like, or 2035, for that matter.

Okay. So they open this report by explaining who they are and what they used as the source of their various Assessments. And they capitalize the "A" of Assessment. So they said: "NCSC Assessment (NCSC-A) is the authoritative voice on the cyber threat to the UK. We combine source information from classified intelligence, industry knowledge, academic material, and open source to provide independent key judgments that inform policy decision-making and improve UK cybersecurity. We work closely with government, industry, and international partners for expert input into our assessments.

"NCSC-A is part of the Professional Heads of Intelligence Assessment." We have another acronym, that's the PHIA, the Professional Heads of Intelligence Assessment.

Leo: The PHIA.

Steve: "PHIA leads the development of the profession through analytical tradecraft, professional standards, and building and sustaining a cross-government community. This

report uses formal" - this is the part I love - "formal probabilistic language," and then they say "(see the yardstick) from NCSC-A product to inform readers about the near-term impact on the cyber threat from AI. To find out more about NCSC-A, please contact the NCSC directly."

Okay, now, the probabilistic language is what's so wonderful here. We have a chart. I've got it in the show notes for anyone who is curious. The chart is labeled "Likelihood of events or developments occurring." And so this is a spectrum. And on the far left we have 0% likelihood of events or developments occurring, all the way to the far right, where we have 100% chance. But we need to apparently precisely define our probabilistic language. So they then ask themselves the question, how likely is a realistic possibility? What does that mean?

Leo: They wasted a lot of time on this.

Steve: Oh, Leo.

Leo: I can imagine the debates back and forth.

Steve: And we've got shades of blue.

Leo: Oh, my god.

Steve: And so, you know...

Leo: What a - oh, geez.

Steve: Yeah. At first this whole effort appears to go right off the rails, since, as I said, they present their so-called "yardstick" which firmly establishes the meaning, with specific percentage ranges, of probability of the various terms their report will use if they ever actually get to reporting.

Leo: It's like debating the shape of the table at the conferences.

Steve: That's right. That's right. And wait a minute, who gets to sit where? Okay. While I'm certainly no lover of bureaucracy, and my first instinct was to balk at this entire thing as make-work, I can see the need to define what "almost certain" means.

Leo: Yeah. They could have saved some time just by putting a percentage in the actual...

Steve: They really could have.

Leo: ...prediction.

Steve: But what fun would that be, Leo? You know. So how almost certain are they? And what about "highly likely"? How likely would that really be? And what's a "realistic possibility"? Those questions are all, you know, nearly answered using the handy yardstick that they provide which shows essentially, as I said, a spectrum of likelihoods ranging from "remote" to "almost certain."

Leo: Except that it's probably a meter stick, not a yardstick. But other than that...

Steve: Oh, that's a good point. Although they actually did use the term "yardstick." So maybe they westernized it.

Leo: These old - they sneak in, don't they, these old...

Steve: Or anglicized it. Anyway, so they have "remote," "highly unlikely," "unlikely," "realistic possibility," "likely or probably," "highly likely," and "almost certain." Now, that's the range. Right?

Leo: Somebody's saying in the Discord, they're missing the "snowball's chance in hell" section. Where does that fit in? Okay.

Steve: So now that we have those...

Leo: Yes.

Steve: ...we'll all know what they're talking about when they make the following judgments based upon all of that data that they've gathered from all of their many primary sources. So here they are. We've got six judgments. Artificial intelligence (AI) will almost certainly continue to make elements of cyber intrusion operations more effective and efficient.

Leo: Yes.

Steve: Leading to an increase in frequency and intensity of cyber threats.

Leo: By the way, since four out of six of their conclusions are "almost certainly," they wasted a lot of time on those other colors [crosstalk].

Steve: Yeah. But Leo, you have to know where you're coming from.

Leo: Oh, my god.

Steve: You have to know your roots, or you wouldn't know how almost certain they were. There will almost certainly be a digital divide between systems keeping pace with AI-enabled threats...

Leo: Yes?

Steve: ...and a large proportion that are more vulnerable, making cybersecurity at scale increasingly important to 2027 and beyond. I'm sorry, cybersecurity app scale, meaning let's get serious, folks.

Leo: This contains no information, this sentence.

Steve: No, no.

Leo: Yes, there will be some systems that are vulnerable, and there will be some that aren't. So we'd better pay attention.

Steve: That's right. Right. But now, number three, assuming a lag...

Leo: How much did they get paid for this?

Steve: Yes. By the word, apparently. Assuming a lag, or no change to cybersecurity mitigations, there is a realistic possibility, thanks to our chart, we know exactly...

Leo: Okay. Wait a minute, I've got to check the chart. When is that? Where does that...

Steve: Where does that fall exactly?

Leo: It's kind of in the middle. Okay? That's 40 to 50%.

Steve: Now, Leo, should I note that the chart has weird gaps? That is...

Leo: Yeah, what is this?

Steve: Like, notice there's nothing between 35% and 40.

Leo: Yeah, nothing is 37% likely. Not.

Steve: Yeah. So in between it being unlikely and a realistic possibility. What happens if something falls in there?

Leo: Yeah.

Steve: I guess that's called "falling through the crack." It's literally a crack in the chart.

Leo: That's called putting too many numbers after the decimal point is what that is. This is - you cannot measure this that accurately. But okay, fine. Go ahead.

Steve: Yes. Precision versus resolution, the two are not the same.

Leo: No.

Steve: "Assuming a lag," they wrote, "or no change to cybersecurity mitigations, there is a realistic possibility of critical systems becoming more vulnerable to advanced threat actors by just two years from now, 2027. Keeping pace with 'frontier AI'" - oh, and by the way, Leo, I left off the glossary at the end, where they clearly define what do we mean when we say "frontier AI." Where exactly on the frontier would that fall? Anyway: "Keeping pace with 'frontier AI' capabilities will almost certainly be critical to cyber resilience for the decade to come." Even though they're only looking two years ahead.

Leo: Yeah. It's a realistic possibility.

Steve: That's right. Oh, no, that's been defined, Leo, as a "realistic possibility."

Leo: I'm sorry, I have to check the averages again.

Steve: Semi, where is that, that's sort of a blue green.

Leo: Oh, that's still in the middle. That's only 40 to 50...

Steve: Oh, no, no, that's more of a green green, yeah.

Leo: Yeah. I'd say it's a higher than realistic possibility. I'd say it's almost a certainty.

Steve: I would agree that's a debatable point. They should go back to their primary sources and see if they don't think that...

Leo: Yeah. Will critical systems become more vulnerable to advanced threat factors in two years? Uh, yeah.

Steve: Uh-huh.

Leo: Okay.

Steve: So there it is. In other words, in the estimation of the United Kingdom's National Cyber Security Centre, that apparently has, like, maybe some excess time on their hands...

Leo: A lot of time on their hands, yeah.

Steve: ...it appears to be "highly likely" that the bad guys are going to be quicker to exploit the many possible nefarious benefits offered by AI than the good guys are going to be able to use that same AI, probably hampered by all of the restrictions we're going to put on it to make sure it doesn't escape, to quickly make today's systems more secure. Now, if that wasn't gloomy enough, they added: "This report builds on NCSC Assessment of near-term impact of AI on cyber threat published in January 2024. It highlights the assessment of the most significant impacts on cyber threat from AI developments between now and 2027. It focuses on the use of AI in cyber intrusion. It does not cover wider threat enabled by AI, such as influence operations. AI and its application to cyber operations is changing fast. Technical surprise is likely."

Leo: Technically.

Steve: And of course "technical surprise" from one's adversaries is never what we want. So overall it appears "almost certain" that it would be a good idea to buckle up, folks.

Leo: Oh, boy.

Steve: We have some interesting times ahead for the industry, and we'll be right here taking a look at everything every week on this podcast as it happens.

Leo: Yes. Almost certainly. But not definitely.

Steve: Almost. It's a high reliability.

Leo: Some probability, yes.

Steve: Because, after all, we have never missed a week. So it's a realistic possibility. I think what we should do, Leo, before we start in our listener feedback...

Leo: Yes.

Steve: ...is remind our listeners...

Leo: I think there's a high likelihood that there's an ad coming up.

Steve: A high likelihood of an ad, yes.

Leo: Is that what you're saying?

Steve: It would be a realistic possibility.

Leo: There's a reasonable certainty.

Steve: Okay. Jim Reed said: "Dear Steve, in SN-1024 you quoted an email from Alex regarding speed test providers online." And he quotes me: "There are dozens and dozens of them, even white-label versions of the most (in)famous, the Ookla speed test." He said: "I've never really trusted the results because most of these are all about ads and the like."

Jim says: "As a former ISP executive, I spent a good deal of my time in speed test discussions." So he's an ISP executive on the other side of all this. He said: "Until 2023, the Ookla service was operated on behalf of the Measuring Broadband America program of the FCC. Using these servers would help provide 'some' accountability back to the ISP." He says: "My ISP maintained a relationship to see aggregated data to help understand how we were doing." So that's kind of cool. They were looking at what their own users were seeing out at the subscriber end in order to get that feedback.

He said: "Here's a big secret you should know. More times than not, it's not the ISP that has the network problem causing slowdowns. It's people with a TV on the patio on the edge of their WiFi coverage going, 'Why can't we see the game?' It's someone speed-testing a gigabit connection with 100 Base-T Ethernet on their machine. It's an eight-year-old iPad that has old generations of WiFi. I could go on, but you understand the issue.

"I'm not exempting the ISP from the discussion because things happen on networks. It only takes a few people to decide they need to download all 700-plus episodes of 'The Simpsons' NOW to cause in-network impacts. The ISP needs to manage for those potentials, and sometimes they just can't see things coming. Everyone starting to work from home during COVID is an example.

"If I had to give advice on speed testing, here's what I would suggest. If you're concerned you're getting what you're paying for, test from a wired connection on your best device. If that doesn't meet expectations, talk to your ISP's support team." He said: "Steve, I've been out of the ISP game for five years now, but I always feel like going back to the basics is a good way to start troubleshooting. Since my retirement I get the latest episode of Security Now! every Wednesday morning and start listening on my walk."

Leo: Nice.

Steve: "Thanks for continuing my computing education far past retirement. Best, Jim Reed." And then he says: "P.S.: 73 to W6TWT from N4BFR."

Leo: That's me, W6TWT.

Steve: I knew the TWiT, the TWT. That's great.

Leo: Tango Whiskey Tango, yeah. And 73 back to you, Jim.

Steve: So anyway, Jim's experience and his observations, you know, matches my own. I've always found that any interruption in my cable modem's connection is due to something on my end rather than something at Cox's end. I'm sure they occasionally need to rearrange things at their end. And when nothing seems to have changed on my end, it would seem obvious that the other guy must be to blame. But almost invariably, when I mess with connections, even those that have been problem-free for years, or I reboot something that appears to be running just fine, the problem will be resolved. And it's always a relief for me since I have far more control over my end than I have over Cox's end. So, you know, I'm happy if it's my end because I can fix my end.

Simon Griffiths wrote: "Hi, Steve. First, thanks for the podcast. I've been listening since the first episode, and it's really helped me in my career and my understanding of IT in general. I control a bunch of different web servers on AWS, AliCloud and others, so I was interested in your discussion of changing the SSH port, which I have done on one server. On AWS and AliCloud you can configure the ports you can use to connect to the server or disable them entirely. What are your thoughts on disabling them completely, then logging in to AWS to re-enable them before you SSH in, or use AWS built-in console?" He says: "I don't actively monitor SSH connection attempts, mostly because I know it would be a bit scary; but your comment on reducing traffic to the site really would be an advantage for almost all web servers. Cheers, Simon."

And I think Simon's approach makes a lot of sense. You know, what we're hoping to eliminate by moving a port out of the main traffic pattern is its overall exposure. So if the option exists to only enable remote access during those times it's needed, by all means take that option. A closed port is even better than an open port that's been relocated to some backwater location.

Jon Borgen wrote: "Hey, Steve. I just got done hearing you talk about Cloudflare's speed test, and wow. It sure is good. I want to share my favorite utility, though, because it's unique in a few ways. It's testmy.net." So T-E-S-T-M-Y dot N-E-T. He said: "What it does that I haven't seen anyone else do is actually upload and download chunks of random data, giving you a much more accurate view of your bandwidth, instead of just a theoretical throughput. You can also schedule a speed test to happen every 15 minutes or hourly if you leave the tab open, so you can get a better understanding of your bandwidth throughout the day. And it runs natively in any browser, including phones and pads. Anyway, I just wanted to share it because it's my favorite. Thanks for all you and Leo do."

Okay. So I just I went over to "testmy.net," and I saw that you had, Leo, also. And yes, it looks nice. You know, it's certainly a consumer-friendly site with some pretty graphics. And the domain "testmy.net" is convenient and easy to remember. The only thing I'll mention is that what all of these bandwidth testing sites are doing is uploading and downloading chunks of data. They don't have any choice. That's the way they work. I can't say for certain whether that data is random. And, in truth, whether the bits are all zeroes or all ones or alternating one and zeroes or random makes absolutely no difference to the outcome.

Just for the sake of argument and illustration, if some form of on-the-fly compression and decompression existed in generic Internet connections, then the content of the data that was being sent and received would matter, since as we know anything encrypted is by definition absolutely uncompressible. But typical data, you know, is not being compressed end-to-end. We don't have compression on by default on Internet links. So anyway, I'm happy to put "testmy.net" on everyone's radar. It looks another useful speed test.

Steve Main said: "Hi, Steve. I wanted to share with you just how wrong ChatGPT can be using the PAID version of ChatGPT 4o. It is such a great tool, and it's still useful; but you have to be very careful as it can be so confident in its answer even if asked 'Are you sure?' It took me PROVING it with a screen shot before it would admit it was 100% wrong and never once double-checked its own work while it told me to double-check my work. I wanted to share this as I really think that this is an important point to drive home to people, to be very careful using LLMs."

He said: "This is why I do not think they will be replacing anything any time soon, and they're just next-generation search engines. I use it for coding PHP and JavaScript, and it is amazing how much it fails so many times. It's great for generating code fast, but it's so buggy, and it just hallucinates functions that don't exist. Again, still faster. But I've now had about 50 screw-ups with it over two months with it on a coding project."

So anyway, I know that Steve Main's experiences, findings, and what he's related echoes many others and mine, as well. I thought it was notable that he referred to using AI as a next-generation Internet search engine since as I mentioned several months ago, that's how I sort of found myself using it, and I know, I just heard you, Leo, and Alex talking about that on MacBreak Weekly, that it arguably has impacted some of Google's direct use because you can, you know, you can get good search answers out of ChatGPT, which is now able to also look at the web, where I guess initially it wasn't.

So anyway, just another bit of feedback on, yes, it's not like we yet have the perfect oracle that's going to give us perfect code every time. I still hold that if anyone actually has an interest in creating a perfect coding AI, and I think there's big bucks in doing that, it really ought to be possible. And it's probably not going to be a generalist that you can also ask it how to tie your shoes.

Lew Wolfgang wrote: "Hi, Steve. In regard to your recent explanation of SSD data retention issues, the thought occurred that it would be nice if just reading from all cells would refresh their charges. I'm reminded of reading from core memory, where the cores have to be flipped to read their polarity, if I remember correctly." And he says: "Yes, I've used core RAM." And, well, Lew, I don't know if you know, you probably do know, that I've used it, too. I recall that we talked about the...

Leo: Used it or seen it? You've actually used it. That's amazing.

Steve: I've used it. I've held it in my hands. And in fact I forgot this was going to happen. I could hold up a - I have it. I own core.

Leo: Oh, yeah I have some in my - I have some, too. I have some framed core memory.

Steve: Yup.

Leo: Yeah. But I never...

Steve: And I remember...

Leo: ...remember having used it. When did you use it? Didn't they have solid state by the time you were an adult?

Steve: No.

Leo: No?

Steve: No, no, no. Those PDP-8/e were - and Leo, they had 4K of core.

Leo: Ooh.

Steve: I mean, 4K words, 4,096 12-bit words because those are 12-bit mini computers. And that was where I learned my first assembly language was programming the PDP-8/e when I was - I think I was a sophomore in high school. And then afterwards, after Berkeley I worked for a company called mini computer technology. We had Nova and Data General machines, and those were all core. So, I mean...

Leo: Amazing.

Steve: ...core was around through the '70s and early '80s because, while there were semiconductor memories, they were still 4,096 bits, and they were like $1,000.

Leo: Right, expensive, yeah.

Steve: For 4,096. So you needed a bunch of those to create words or bytes. And it was just so expensive back then. And the density was still so low.

Leo: You'd think core would be more expensive. Somebody had to wire all that stuff. But okay.

Steve: Yeah. Yeah.

Leo: Wow.

Steve: So Lew is correct. Core memory used a technology known as "destructive read" because the process of reading a memory location inherently destroyed what was originally stored there. The way it worked was that the reason it was called "core memory" is that these little, as they called them, "cores" were circular ferromagnetic

rings that had wires threaded through them. And being a closed ring, the ring could be magnetized in either a clockwise or counter-clockwise direction. And a "sense" wire, as it was called, would run through the center of the ring. It was able to sense when a - that's a beautiful picture - when a ring switched its direction of magnetization since this switching event would induce a pulse in the so-called sense wire running through the ring, and that pulse would be picked up by a sense amp.

So the process of reading out a location of data from core memory required all of the rings representing the bits of that memory location be written in the "0" direction, that is, be pulsed so they all switched to the "0" direction. Now, when that happened, only those rings that were originally set in the "1" direction would be reversed to the "0" direction, and that reversal event would produce a pulse on their respective bit sense lines. This allowed the computer to determine what had been stored in that location. Now, and Leo, so what you're seeing is that red wire, that red wire which loops back and forth over and over.

Leo: It crisscrosses through it, yeah.

Steve: Yeah. That is the sense line for that little chunk of memory, and you can see that it's actually a continuous loop. Two wires come out from the left, and then that wire zips back and forth, passing through each ring exactly once, and then goes back out to the left.

Leo: By the way, they never were able to manufacture these with machines.

Steve: No.

Leo: People actually threaded them. Hand-threaded them.

Steve: Hand-threaded those cores. And boy, they got very, very tiny over time.

Leo: Yeah.

Steve: Because, you know, again, density always wants to go up.

Leo: This is a 128-byte core memory. This is from Wikipedia.

Steve: Very, very cool.

Leo: Yeah, yeah.

Steve: So anyway, in order to determine what was in there, you had to write, you wrote all the cores of one location to zero. So any of the cores that had been set to a "1" direction were forced to reverse direction, and it was that direction reversal that induced a pulse in the sense line that then was able to be picked up. So now you know what was

stored in the memory. But unfortunately, you've just erased it. You've just set it to all zeroes in the process.

So if the instruction you were executing was, for example, a load, loading what's in memory into a register, then you would need to immediately rewrite what was read from it back into it. And so it would be a read/write cycle. But what was really cool is what if we wished to increment a binary value stored in a memory location like that? You know, to add one to it. So the clever computer designers of the era realized that this could be accomplished at the same time as that necessary rewrite of the original data for that location. It was known as a "read-modify-write" cycle, where after reading a value from core memory, the value that had just been read would be passed through an adder to add or maybe subtract a one from it, if you wanted to decrement the value in memory, and that modified value would be what was rewritten into the core memory instead of replacing the location's original value.

So a core memory's "cycle time," which was the time it took to read and rewrite or read, modify, and write, that was, it turned out, that was the slow part of computers of that era. The electronical part they could do really fast compared to the actual physics of ramping up the current, giving the core time to switch its field, and then ramping the current back down, capturing those pulses, seeing whether they exceeded a minimum peak value, deciding that it was a one, all of that took time. So it was the memory cycle time that determined the rate at which instructions could be executed.

So anyway, Lew's point was a good one. If reading from SSDs could refresh the charges stored in their cells, then a simple read pass could be used to keep the bits firmly written. Unfortunately, as we know, that's not the way today's NAND-style flash memories operate. And frankly, given that memory is in general read much more frequently than it's written, if we had to choose which process, reading or writing, would cause cell fatigue, it's way better the way it is, where the least frequently performed operation, which is to say writing, is the one that takes a lot more time and then also produces the wear on the cells.

Leo: I wonder what the proportion of reading to writing is on average. It's probably 10 to one, at least; right?

Steve: Oh, I would guess it's more like 100 to one.

Leo: Yeah, yeah.

Steve: I think it's very high.

Leo: You don't change data that often.

Steve: Nope. Nope. And that's why we see it slowing down is that it just - the cells are never or almost never being rewritten, unless there's some reason to do that.

Leo: Right.

Steve: Chris said: "Hey, I was just listening to this episode and the listener story about Verizon" - oh, no, it was about "Horizon."

Leo: Horizon, yeah.

Steve: Yes, "about Horizon was interesting."

Leo: Yeah.

Steve: He said: "About six months ago, I wanted to try their wireless gateway. I went to a local store, and they would not sell me anything until I unfroze my credit and let them run a credit check."

Leo: Yeah, that's normal.

Steve: Yeah, exactly. He said: "I'm not sure if that listener had done that, or somehow the bad guy weaseled around it, but I tell anyone who will listen that they're crazy not to freeze their credit given the protections it gives you 'in the real world,' too. Enjoy the show. Chris."

So I figured I'd just share Chris' experience for the sake of another viewpoint. You know, given that the criminal purchaser presumably walked out of the store with a brand new multi-thousand dollar smartphone, and that their "creditworthiness" would seem to be a huge factor in that, it is indeed puzzling how this happened to the person who wrote to me before. You know, perhaps it was just vendor error at the "Horizon" store.

Leo: Yeah. Or, you know, remember customer service reps try to be friendly and nice.

Steve: Yeah.

Leo: If you can persuade them to bypass their normal protections, you know, a lot of hackers are good at that.

Steve: And they probably get, I'm sure they get credit for selling an expensive phone.

Leo: Yeah, that's true, too.

Steve: I wonder if they get dinged when it doesn't get paid for. Lee MacKinnell said: "I decided to pull the plug on my Microsoft password as I already had passkeys set up." He said: "Microsoft makes you jump through some bizarre hoops. First, install the Microsoft authenticator TOTP" - you know, time-based authenticator app on my device. He said: "It lets you use other password apps for TOTP code verification, so I set it up in Bitwarden. But you need THEIR app to turn off passwords." He said: "Will get to that soon. Second, verify you have the TOTP app installed by providing a generated code as verification.

Turn on passwordless login for microsoft.com. And then, fourth, you are then prompted to open Microsoft's authenticator app to verify you want to disable passwords on your account." Then he said: "Yay. I am now passwordless."

He says: "Now, logging in passwordless. First, enter your email address. Second, you're prompted to verify login with the Microsoft Authenticator app, asked to select the matching two-digit code in the app." That was interesting. I don't know what that meant. I'm just reading what he shared. He said: "At this point I selected 'Other ways to sign in' because I have passkeys." Then he says: "Shouldn't this be the default?" Then: "Three, the first option is now 'Use your face, fingerprint, PIN, or security key.'" And then: "Fourth, the passkey process is then started so I can log in." And then he said, he finished: "At this point I am unable to remove the TOTP login option from my account or set passkeys as my default login option."

So it sounds a little screwy to me. I haven't tried doing any of this myself yet, so I'm now as confused as Lee sounds. I don't dare mess up my original Microsoft account since it's tied to my access to developer tools and downloads. So what I may do is just create a new account, a separate account for testing. But I have not done so yet. I imagine I'll hear from other listeners who will help us figure out what's going on.

Leo: Yeah.

Steve: Assuming that Lee is correct, it's annoying that Microsoft appears to force the use of their own authenticator app, though I'm not that surprised, since, as we know, all time-based one-time password apps should be equal. And in fact, allowing the user to use the one they prefer to use makes the most sense, you know, makes much more sense than forcing them to use, you know, a specific app. That's certainly not in keeping with the way we would expect it to work.

Leo: I think what it's doing is different from the - I don't want to show this. It's an eight-digit number.

Steve: Oh.

Leo: I did disable my password in Microsoft. I did not find it difficult. But I also use - I don't use the Microsoft Authenticator for anything but Microsoft.

Steve: Right.

Leo: But I've been using that for a long time, like was logging into my Microsoft account. You know, it says on your computer a two-digit number, and then you have to confirm the two-digit number in the authenticator. You've done that; right?

Steve: I've not...

Leo: No.

Steve: ...used Microsoft's Authenticator.

Leo: Oh, okay. Well, I'm not surprised that it, I mean, I had - but since I use it, it was a very simple thing to disable a password.

Steve: Okay.

Leo: I had some, you know, I was a little nervous about it. But you told me it's better not to have any password.

Steve: It is better not to. So you were able to delete your password.

Leo: Oh, yeah, yeah.

Steve: Very cool.

Leo: It says right now, I don't want - again, I don't want to show you.

Steve: No, no.

Leo: But it says "This account does not have a password."

Steve: Nice.

Leo: You can use this device to sign in instead.

Steve: Nice. Okay. So this is not just a TOTP, then. It is a...

Leo: It's a passkey [crosstalk].

Steve: It is a passkey authenticator.

Leo: It's single sign-on.

Steve: Okay.

Leo: Yeah. So it's not a six-digit TOTP. It's a...

Steve: That actually - that resolves our confusion.

Leo: Yeah, yeah. But it works great. I'm very happy with it.

Steve: Nice.

Leo: Yeah, yeah.

Steve: Bienvenido Del Rosario said: "Dear Steve, I'm a long-time listener and Club TWiT member from the Dominican Republic."

Leo: Yay. Thank you.

Steve: Yes. "I wanted to share my experience with an open SSH port on my home server after hearing your recount of the many authentication attempts Gaelin was receiving on the Security Now! Podcast Episode 1023. I was using public/private keys" - good, best way - "for my own login" - get this, Leo - "but I had between 70,000 and 80,000" - eight zero zero zero zero, 80,000 - "daily failed authentication attempts."

Leo: Yeesh. That's somebody hammering you. Wow.

Steve: Yeah. "After installing and configuring Fail2Ban, the attempts went down to around 20,000 per day. Even though it was a big reduction in failed attempts, I still was feeling very concerned. I started reading posts about how to mitigate even more of the failed attempts and landed on a very simple solution: Change the SSH port from the default of 22 to any other number in the user port range, from 1024 to 49151. And voila, all failed attempts ceased right away. Since then, I resorted to just closing the open custom port, and I use Tailscale to access my home lab, and I've been very happy ever since. I hope you find this information helpful. Best regards, Bienvenido Del Rosario." So...

Leo: Just another day of corrected. I'm going to change the port.

Steve: Yup. Makes sense to get it out of the main trajectory. All the incoming missiles are aimed at port 22. Scott Schabel said: "Hi. The recent discussions about the use of non-standard ports for services that don't need discovery validates my use of the practice for years, so thanks. The latest Security Now! discussion got me wondering, what about a non-standard port on an IPv6 address? I can't imagine how it would be easy to scan and find it - again, not for security, but for 'Why Not?'" He said: "I still have IPv6 turned off on my network, but am wondering if I should start working to turn it on. Thanks. Scott."

So it's true that moving to IPv6 would dramatically increase the address space the bad guys would need to scan. Although only a fraction of the total 128-bit IPv6 space has been allocated to the world's ISPs, it's still true that the world's ISPs now have vast IP space for their customers. And like Scott, I had also been routinely disabling IPv6 on my networks, since being old-school I saw no need for it. But my work on the IPv6-capable DNS Benchmark code required that I have IPv6 IPs and IPv6 protocol up and running,

and it's been working without any trouble, both here on my Win7 box and on my Win10 box in my other location.

That said, among those who have been testing the early pre-release Benchmark code, there are many whose ISPs are not offering IPv6 and appear to have no plans to do so. So I've needed to have the Benchmark accommodate its users who do not have access to IPv6. They'll still have IPv4, DNS over TLS, and DNS over HTTPS, just not IPv6. And after the Benchmark verifies that they need to skip IPv6, it won't bother them about it. But until they do it says, hey, you don't have IPv6 working. Do you know that? Do you want to? Should you? Anyway, so yes, you can certainly play with IPv6. It works.

And I've got a great note from a contributor and friend online named Greg Bell, Leo. Let's take a break, and then I'm going to share what someone who calls himself "ferrix" in our groups has to share.

Leo: I think I'm familiar with ferrix.

Steve: So Greg said: "Hi Steve, Greg (ferrix) here under a different email due to the mailing list. Regarding Windows 11 vertical taskbar, I thought you might enjoy knowing there's a bit of a story on this one. Matching your intuition, there are some tools that give you back the Windows 10 taskbar, or something like it, on Windows 11. Some work by turning back on code that MS has disabled and may at a time delete, or by being shell replacements. Stardock and StartAllBack are examples."

He said: "I wasn't satisfied with that because I'm accustomed to my taskbar behaving precisely how I want it to. Since Windows 7 through 10, I have run the famously useful '7+ Taskbar Tweaker' utility." He said: "It improves all kinds of dumb behavior and limitations in the Windows taskbar management experience, without replacing the taskbar or Explorer. Over a year ago, when I determined that MS would not support the only sensible vertical taskbar in Windows 11, I looked to Michael, the author of the above taskbar tweaker tool. He's been improving my taskbar experience ever since Windows 7. Could he vert-ify the Windows 11 taskbar?

"I learned that Michael now makes a generalized system called 'Windhawk' [W-I-N-D-H-A-W-K] to inject various user-selected changes into the Windows UI experience. Instead of a monolith with a million checkbox features like the old tool, this is more of a framework that runs only the plugins, a.k.a. 'mods,' that each user wants. It doesn't replace Explorer or any other process, but instead just inserts clever little hooks here and there to make Windows do its bidding.

"Michael looked into the vertical taskbar issue and despaired. Windows 11 taskbar is an almost complete rewrite. Turning on verticality in the new bar cannot simply be done since it was never implemented by MS in the first place. It would take at least weeks, if not a couple months, of development to build such a feature into the Windhawk system, if it was even possible at all, and more time than Michael could afford to spend on such a hobby project. So that's where he left it.

"But my company small development staff, including mostly me, rely on a vertical taskbar, and it would be a massive efficiency hit to lose it. So we contracted with Michael to build the feature for us. And although we paid for the initial development, we don't own it. We agreed that the feature should be freely available to everyone, just like Windhawk itself." He says in parens, "(Not for nothing, having a bunch of other users running this also provides feedback to make it better over time, a concept I know you're well familiar with, with your own product development strategy). In any case, I present

Vertical Taskbar for Windows 11." And I've got a link in the show notes. And, he said, "and a somewhat dated Reddit thread about it," and I have a link there, too.

He said: "I don't get anything" - yes, Leo, look at that. He said: "I don't get anything for touting this tool, and I know the idea of running such an extension may not be universally appealing. But I thought at least you'd find it an interesting tale. I think you, Michael, and I share a certain perspective. We don't write operating systems or build the SSDs. Windows didn't plan on Windhawk being there. Active Directory didn't understand YubiKeys on its own." Which, by the way, is what Greg added. "Hard drives aren't built to expect SpinRite or ReadSpeed coming along. But with a little leverage in the right place, we can make other people's systems work better than their original design parameters."

Leo: This is - I'm sorry, go ahead, I didn't mean to interrupt.

Steve: And he said: "And there's no feeling quite like it."

Leo: This is an amazing tool.

Steve: It is, Leo.

Leo: Holy camoly. And it's free?

Steve: Yes. It is an amazing piece of work. So I perked up when I saw email from Greg in the first place. I know him quite well from his many years of involvement and contributions to GRC's various online forums. And this "Windhawk" system which he brings to our attention is truly amazing. So go to Windhawk, W-I-N-D-H-A-W-K dot net, Windhawk.net. Then at the top of the page click "Browse for Mods," which is what Leo did, or go to Windhawk.net/mods. By default, the page is sorted from the most popular/most installed to least, with the most having nearly 143,000 users, and the least having two. And oh, my god, I have no idea...

Leo: What a gift this is. This is amazing.

Steve: ...how many mods there are overall, but the mod list page scrolls and scrolls and scrolls nearly without end. It's got to be hundreds and hundreds, and they are very specific mod tweaks. So, for example...

Leo: It's not just taskbar, it's everything. I mean, wow.

Steve: Yup. So, for example, reading from the most popular, Windows 11 Start Menu Styler is a mod. Customize the start menu with themes contributed by others, or create your own. Then we have the Windows 11 Taskbar Styler. Customize the taskbar with themes contributed by others or create your own. The Taskbar height and icon size: Control the taskbar height and icon size, improve icon quality for Windows 11 only. Or - and by the way, all this is Windows 10 unless it says Windows 11. Windows 11, oh, and probably Windows 7, too. Windows 11 Notification Center Styler: Customize the

Notification Center with themes contributed by others or create your own. Taskbar Volume Control: Control the system volume by scrolling over the taskbar.

Better file sizes in Explorer details: Optional improvements: show folder sizes, use MB and GB for large files rather than always being stuck on KB and also, if you want, use the IEC terms, you know, KiB instead of KB. We have Taskbar Clock Customization: Customize the taskbar clock. Define a custom date/time format, add a news feed, customize fonts and colors, and more. Slick Window Arrangement: Make window arrangement more slick and pleasant with a sliding animation and snapping. Taskbar Labels for Windows 11: Customize text labels - well, I can go on and on and on because there's literally hundreds of these little tiny mods.

Leo: So how does it do this? Is it - are they registry?

Steve: And Leo, did I fail to mention that all of the source code is provided?

Leo: Oh, wow.

Steve: If you click on the details for any mod, the tabs are "Details," "Source Code," and "Changelog." So as Greg said, these are all going to compile down to very tiny modules. Because you can click on source code and look at the source code of these modules to see how they work.

Leo: Look like they're written in C.

Steve: Yup.

Leo: That's interesting.

Steve: The "Mods" page can be sorted many ways, and it has an incredibly fast and responsive incremental search. Since I was curious about the vertical taskbar Greg's company commissioned, I entered V-E-R-T and was looking at "Vertical Taskbar for Windows 11," which, you know, describes itself as "Finally, the missing vertical taskbar option for Windows 11." And I put a picture of it in the show notes. I mean, it looks like exactly what you want.

Leo: Yup.

Steve: There it is.

Leo: So this, you used to be able to drag the taskbar around. They turned that off for Windows 11?

Steve: You never had that in Windows 11. It's always been in all previous Windows. And Windows 11 they just unilaterally decided, nope.

Leo: This is it.

Steve: You're going to have it in the - we're going to - what you can do now is at least you can have it float over to the left. So left alignment they have condescended to. But not verticality. There's none of that. So now we have it.

Leo: And you still use Windows; huh?

Steve: Leo, that's where everybody is.

Leo: I know, you have to.

Steve: I write apps for the majority desktop, and it's all Windows.

Leo: Yeah. Wow. It's really interesting that you can write code that will modify the operating system this dramatically. I find that fascinating.

Steve: Yeah, modify the UI.

Leo: Yeah. Well, good on Windhawk.

Steve: So Greg, thank you very much. And I have no doubt that Windhawk.net is going to get some traffic from people saying, oh, I want to see what I can do, and then doing it.

Leo: There's a lot of stuff.

Steve: And doing a really nice job of it. I love the idea of it being just a little framework, and then you just, you know, little individual modules that do the things you want them to.

Leo: Hey, before we talk about secure conversation records retention, and your invention, which I'm very excited to talk about - Steve has solved the problem, and he will tell you his solution. So everybody listen.

Steve: I think I probably do have it, and I'm hoping that it gets, I mean, now I know from our listeners that a bunch of people want this, too, and it's not a hard solution. So I hope people get on it.

Leo: Awesome. All right, Steve. I am just dying to hear your solution to all of this.

Steve: Okay. So what is end-to-end encryption? And what does it mean in environments with requirements for the long-term retention of records? As civil disputes have arisen in an information age, attorneys have sought to obtain records of prior events that may not have been retained. The result of this has been a growing requirement for records retention articulated by laws such as the Federal Records Act, the Freedom of Information Act, the U.S. Presidential Records Act, and the Federal Rules of Civil Procedure. And we've encountered many instances where private companies are required by law to retain their own records in the event of litigation.

The recent events surrounding Signal and TeleMessage and members' of the U.S. government's use of end-to-end encryption, with TeleMessage's mission to archive conversations, raises many questions about the intersection of secure communications and the need for long-term records retention.

Okay. But I'm getting a little bit ahead of myself. I settled upon this topic, as I said at the top of the show, during this week's podcast only after catching up with the news of an event that was the topic of last week's podcast. Because it was not my original intention to give this whole TeleMessage Signalgate story much more air, I originally had this as a news item near the top of the podcast. But we've learned some more in the past week, and the news legitimately and ultimately led me to pose the question that became today's topic. So let's all catch up on what transpired over the past week.

Under the summary line "Three U.S. Departments Ban TeleMessage," the Risky Business security newsletter wrote: "According to Bloomberg, three U.S. government departments have told employees to stop using the TeleMessage service. The service allows companies to log and archive conversations taking place in secure messengers such as Signal, WhatsApp, and others. Two separate hackers" - two - "breached TeleMessage's backend last week after it was revealed that White House officials use the service."

Now, WIRED had the headline "Customs and Border Protection Confirms Its Use of Hacked Signal Clone TeleMessage" with the subhead "CBP says it has 'disabled' its use of TeleMessage following reports that the app, which has not cleared U.S. government's risk assessment program, was hacked." Rather than share the entire article, which is padded with a bunch of stuff that we already know, I'll extract things we didn't already have on the record.

One, "The United States Customs and Border Protection agency" - so this is WIRED reporting - "confirmed on Wednesday that it uses at least one communication app made by the service TeleMessage, which creates clones of popular apps like Signal and WhatsApp with the addition of an archiving mechanism for compliance with records-retention rules. The CBP spokesperson Rhonda Lawson told WIRED: 'Following the detection of a cyber incident, CBP immediately disabled TeleMessage as a precautionary measure. The investigation into the scope of the breach is ongoing.'"

Okay. So we have confirmation of the belief which arose from the hacked data that was anonymously shared, which we talked about last week, with 404 Media, that the CBP was indeed using the TeleMessage app.

Also, says WIRED: "In the days since the photo was published, TeleMessage has reportedly suffered a series of breaches that illustrate concerning security flaws. Analysis of the app's Android source code also appears to indicate fundamental flaws in the service's security scheme. As these findings emerged, TeleMessage - an Israeli company that completed an acquisition last year by the U.S.-based company Smarsh - imposed a service pause on its products pending investigation.

"A Smarsh spokesperson told WIRED in a statement: 'TeleMessage is investigating a potential security incident. Upon detection, we acted quickly to contain it and engaged an

external cyber' - how can you have a potential incident which you contained? Well, anyway - 'engaged an external cybersecurity firm to support our investigation. Out of an abundance of caution, all TeleMessage services have been temporarily suspended. All other Smarsh products and services remain fully operational.' And 'There is still no complete public accounting of U.S. government officials and agencies that have used the software.'" So we're flying a little bit blind on that side.

Jumping to the bottom of other reporting by NBC News, we find: "But archives of sensitive information inherently make targets for hackers. On Sunday evening, a hacker credibly claimed to NBC News to have broken into a centralized TeleMessage server and downloaded a large cache of files. As evidence, the hacker provided a screenshot of TeleMessage's contact list of employees at the cryptocurrency broker Coinbase, which uses TeleMessage. A Coinbase spokesperson confirmed to NBC News that the screen grab was authentic, but stressed that Coinbase had not been hacked and that none of its customers' data had been affected.

"The Coinbase spokesperson said: 'At this time, there is no evidence any sensitive Coinbase customer information was accessed or that any customer accounts are at risk, since Coinbase does not use this tool to share passwords, seed phrases, or other data needed to access accounts.'

"The hacker told NBC News they have not fully sifted through the hacked files yet, and it is unclear if they include sensitive conversations from the U.S. government. Several government agencies, including the Department of Homeland Security, the Department of Health and Human Services, the Treasury Department, and the U.S. International Development Finance Corp. appear to have active contracts with TeleMessage or other companies to use TeleMessage's services, according to government records reviewed by NBC News. Separately, a different hacker told the tech news publication 404 Media that they also hacked TeleMessage and provided significant evidence. NBC News has not interacted with that hacker."

Okay. So confirmation of more apparent departments inside the U.S. government using TeleMessage and lots of hacking of TeleMessage, multiple confirmations from different sources and directions.

Last week, security researcher Micah Lee blogged under the headline "Despite misleading marketing, Israeli company TeleMessage, used by Trump officials, can access plaintext chat logs." Micah wrote: "In this post I will give a high-level overview of how the TeleMessage fake Signal app" - and I would use the word "clone," but he's using some strong language here - "Signal app called TM SGNL" - because, I mean, it's not fake. No one believes it's Signal. It calls itself TM SGNL, right, S-G-N-L, how it works and why it's so insecure. "Then I give a thorough analysis of the source code for TM SGNL's Android app, and what led me to conclude that TeleMessage can access plaintext chat logs. Finally, I back up my analysis with as-yet-unpublished details about the hack of TeleMessage."

Okay. So among other things, Micah created a clear and simple diagram that depicts the flow of information for anyone using TeleMessage's modified Signal app. The modified app does what it claims to do. But since the archived messages are themselves stored and forwarded in the clear, that does also serve to verify that TeleMessage themselves would, indeed, be privy to the content of the message flow of any of their customers. That's pretty much sure to be a permanent deal breaker for many of the more security-sensitive users of this system.

Leo: Isn't TeleMessage an Israeli company?

Steve: Yes.

Leo: Okay. Just checking.

Steve: Yup.

Leo: So, good, our U.S. government is basically passing all of their messages to the Israeli government.

Steve: That's correct.

Leo: Or the Israeli company, anyway, yeah.

Steve: That's correct. And they are users of TeleMessage. And but wait, Leo. It gets worse.

Leo: Oh, good.

Steve: There's something I haven't read or seen anywhere. This would be totally obvious to anyone with any operational cybersecurity understanding at all. So let me repeat this: The TeleMessage system is depositing the full plaintext transcripts of all conversations conducted with the Signal protocol whenever using this app. My point is, the entire design of this system is so transparently insecure that any claim to security would be utterly laughable. Micah starts out saying "Despite misleading marketing..." Misleading marketing? It's sending everything you sent and received to Microsoft Outlook or any SMTP email account in the clear. That's what it does. It emails.

Leo: It's emailing it?

Steve: Yes.

Leo: In the clear?

Steve: Yes. It uses email.

Leo: Oh my god.

Steve: So there's nothing about this that is secure for the...

Leo: Why bother using Signal if you're going to use this?

Steve: Yeah. There's nothing about this that's secure for the content of these messages. Nothing.

One of our listeners, responding to my reporting of this last week, apparently feeling the need to defend the current administration, claimed that TeleMessage's use had been approved by the Biden administration. At the time I knew nothing either way. Now I do. I presume this listener picked up this politically partisan fiction somewhere and wanted to believe it. But it was never true. And these sorts of statements, expressed as fact, tend to spread, and they're not helpful. The Biden administration may have well done stupid things, but this was not among them. TeleMessage's federal use has never been approved for use by anyone within the federal government. Part of the reporting about this in the past week noted that, while the TeleMessage company is itself a federal contractor, the consumer apps it offers are not approved for use under the U.S. government's Federal Risk and Authorization Management Program, known as FedRAMP. That's the approval that would be required.

And that's actually a relief, that is, that it has never been approved because TeleMessage's apps, which send, as I said, conversation plaintext to any email servers specified by their users, could never have possibly been considered safe or secure to use. This would have to be so blatantly obvious to anyone with even the slightest cybersecurity training that I would be quite worried if these apps could have ever been approved for use under the FedRAMP program. So the use of these communications applications by federal government officials was entirely illicit. So we can put that one to rest.

Micah's blog posting digs deeply into the entrails of TeleMessage's Android app, which is open source. I scanned through Micah's detailed posting and his reverse engineering and did not find anything that merited further deep discussion. But I've included its link in the show notes for anyone who might be interested in looking for more. It's all there, and it's long and detailed.

There's really nothing anyone with any training needs to know once it's understood that TeleMessage is emailing its users' conversation logs to external email servers. From that point on, it's simply "game over." And it must have made those within the NSA and CIA and other security-aware agencies who know how much our adversaries would love to get their hands on these conversations, just, you know, just green.

Leo: Well, guess what? They already do.

Steve: Yeah. This brings us to the first part of the meat of today's topic. When we use the term "Secure Conversation," and where TeleMessage quite clearly failed, what really is end-to-end encryption? And by that I mean, what do we mean when we say that something is end-to-end encrypted? How is that some sort of special type of encryption? Or a special type of encrypted system? And the bigger point is: How is that term increasingly being used, misused, and in some cases abused?

The difficulty, and we should say that TeleMessage absolutely touted their technology as end-to-end encrypted, that's what they said in all of their marketing. And apparently no one looked any further. Nobody who was going to use it thought to ask somebody with any cybersecurity awareness would this be safe for me to use. Because any of our listeners would go, "Oh, my god, no." Okay. So the difficulty is that the term's common use appears to be diverging from its original technical meaning, and has become a buzzword term. Like everybody wants to have, oh, full end-to-end encryption. So it's being tossed around now by the marketing types because it feels good and fancy to say it.

Okay. So let's first turn to the Internet's encyclopedia to see what those who have spent time working to craft a clear and concise definition of the term have come up with. Of end-to-end encryption, Wikipedia writes: "End-to-end encryption (E2EE) is a method of implementing a secure communication system where only communicating users can participate. No one else, including the system provider, telecom providers, Internet providers, or malicious actors can access the cryptographic keys needed to read or send messages.

"End-to-end encryption prevents data from being read or secretly modified, except by the true sender and intended recipients. Frequently, the messages are relayed from the sender to the recipients by a service provider. However, messages are encrypted by the sender; and no third party, including the service provider, has the means to decrypt them. The recipients retrieve the encrypted messages and decrypt them independently. Since third parties cannot decrypt the data being communicated or stored, services that provide end-to-end encryption are better at protecting user data when they are affected by data breaches. Such services are also unable to share user data with government authorities, domestic or international."

Okay. So I'd say that's a beautifully crafted definition of what is currently meant by the proper use of the term. And obviously TeleMessage completely fails in that regard. So by way of comparison to non-end-to-end encrypted systems, the Wikipedia article actually says: "In many non-end-to-end encrypted messaging systems, including email and many chat networks, messages pass through intermediaries and are stored by a third-party service provider, from which they are retrieved by the recipient. Even if the messages are encrypted, they're only encrypted in transit, and are thus accessible by the service provider. Server-side disk encryption is also distinct from end-to-end encryption because it does not prevent the service provider from viewing the information, as they have the encryption keys and can simply decrypt it.

"The lack of end-to-end encryption can allow service providers to easily provide search and other features, or to scan for illegal and unacceptable content. However, it also means that content can be read by anyone who has access to the data stored by the service provider, by design or via a backdoor. This can be a concern in many systems where privacy is important, such as in governmental and military communications, financial transactions, and when sensitive information such as health and biometric data are sent. If this content were shared without end-to-end encryption, a malicious actor or adversarial government could obtain it through unauthorized access or subpoenas targeted at the service provider. Finally, end-to-end encryption alone does not guarantee privacy or security. For example, data may be held unencrypted on the user's own device, or be accessible via their own app, if their login is compromised."

Okay. So one of this podcast's early terms and acronyms was TNO, which stood for "Trust No One." We used it when talking about storing our data in the cloud well before anything was being called "The Cloud." The simple idea was that client-side encryption and decryption would be employed on the user's PC so that the only thing we were asking remote services to store was big blobs of data that were indistinguishable to them from completely pseudorandom data, which is exactly what good encryption produces, pseudorandom data. Later, the term was used interchangeably with PIE - P-I-E - which was another term we used which stood for Pre-Internet Encryption.

So what emerges here is a very clear understanding of what the proper use of the term would be. In practical terms it means that the providers of the service and any intermediaries that they may engage, never under any circumstances have any access to the unencrypted content of the messages or data or whatever it is they're conveying or storing on behalf of their users. For this assertion to hold, any of the user-data that a provider of end-to-end encrypted services ever comes into contact with must be encrypted; and the provider must never, at any time, have access to the cryptographic

keys that would be capable of decrypting that data. By that definition and the set of requirements that flow from it, it's clear that the message archiving services TeleMessage was offering could never - literally by definition - have ever been considered to be end-to-end encrypted. That term could never be accurately applied.

By comparison, we've previously explored at length the protocols that the likes of Apple and Signal have gone to in order to truly meet and live up to the definition of "end to end encrypted" communications and storage. This entire battle that Apple is presently engaged in with the UK over the provisioning of their Advanced Data Protection amounts to exactly this. The switch on the user's UI on the phone is labeled "Enable Advanced Data Protection," but it could just as accurately be labeled "Enable End-to-End Encryption."

It might next be instructive to ask: Is there any hope for TeleMessage? Is there any way for a potentially useful service such as this to be saved? Or phrased another way: "Could message archiving of a true end-to-end encryption system such as Signal, WhatsApp, or Telegram remain end-to-end encrypted?" And the answer to that is yes. There's definitely a way to do this securely, but TeleMessage never bothered to. They opted for the user convenience of forwarding the user's plaintext conversation logs to email, and in doing so they probably put themselves out of business forever. I doubt anyone would ever consider trusting TeleMessage again. So when someone does produce a true end-to-end encrypted messaging system with end-to-end encrypted long-term archiving, it won't be these guys.

Okay. So how could what TeleMessage wanted to do be accomplished securely? The most obvious solution is to simply modify a cloned Signal app for long-term local archival storage. So it would offer local storage, search, and retrieval on the client-side device itself. After all, the plaintext data is already there. It's been shown to its user right there on the screen. So the app simply needs to hold onto it; right? Problem solved. The behavior everyone is objecting to is having these modified TeleMessage Signal clients emailing their conversation logs to its user's insecure email. So remove that feature from the app and replace it with long-term archival storage of everything that app sends and receives.

And to be a responsible archiving Signal clone, any such clone which has been configured for long-term message archiving should periodically inject a notice to all other members of any archiving conversation that its user, "member X," is permanently archiving their conversation. That might be expected to alter the behavior of the people in the group, causing it to be a little bit less boisterous and flippant, but that would probably be a good thing, too. And if nothing else, these periodically injected archiving reminders would cause its recipients to ask the archivist why they were keeping the conversation, if that had not been disclosed.

But now we have a new problem. The problem is that high-level government or other individuals who are using an archiving system, you know, an archiving secure messaging clone, whose entire messaging history exists inside that device, wind up carrying around something in their pocket that might well be worth a great deal to the right parties. No one in their right mind would want to have it known that they're walking around town, riding in cabs and dining in restaurants with months or years of top secret records retained in their smartphones. So that's not a practical solution either.

And even if walking around with this conversation archive was not a bit problematic in itself, the presumption is that aside from having the convenience of searching for and retrieving past conversation details, there is also some significant need for government records to be retained or other executive compliance requirements in the corporate world to be met. If a device were to be lost or stolen, and even if it could not be unlocked and used, the loss of those records could be a huge problem.

This brings us to the inescapable conclusion that the app's local conversation memory should be kept short and non-archival, and that some secure means of removing the data from the device, while retaining it permanently, should be found. The solution I've come up with that completely solves all of these various problems, providing truly secure records retention for users of Signal and using only the official unadulterated Signal app to deliver full true end-to-end encrypted conversation security, would be for these individuals to add a secure government Signal-Bot to their conversations.

This Signal-Bot would be running deep inside a secure NSA facility. It would auto-accept all conversation invitations from known government staffers, and it would passively receive and permanently archive all dialogue from everyone else who was participating in the conversations. It would function as a silent observer and would appear as a participant named something like "Federal Records Retention" in the list of conversation members so that everyone participating would be informed and reminded that their conversation was subject to legally required retention, while at the same time being assured that their conversation was being retained and that they were, thereby, abiding by their oaths of office and the law while participating in these conversations.

Being a Signal-Bot, all of the standard end-to-end encryption guarantees offered by the Signal protocol would apply. The only possible points of vulnerability would be at the individual device endpoints, but that's inherent in the use of any messaging application.

Since Signal is open source, and desktop clients already exist, the creation of various secure message archiving bots would be a simple matter for the likes of the NSA, CIA, CISA, or whomever. Or better yet, perhaps a successor company to TeleMessage will arise from the ashes of TeleMessage's clearly useful and important concept to create a commercial implementation of this solution for those who need Signal message archiving. I would imagine that many enterprises would welcome the ability to automatically have their executives' and other important individuals' secure Signal conversations securely archived on-premises without creating the sort of serious security weakness we've all just witnessed from TeleMessage.

In retrospect, this seems like such an obvious solution for the secure archiving of secure messaging that I'm surprised such a service doesn't already exist. Let's hope someone creates such a system soon. It should not take that much effort or time, and it should definitely remain open source since it's all about first understanding and then trusting the implementation.

Leo: Well, actually, so there is a protocol for this in the federal government. When you're doing classified communications, you're supposed to do it in a SCIF. And the SCIF has logging and recording, and it stays in the SCIF. It stays in a secure environment. So it's very similar to what you just described. And you're not supposed to do classified conversations outside a SCIF at all. So, but yeah, I think what you propose makes sense, or even maybe that, if you're going to do this, you're actually logged into a server inside the Pentagon and interacting only within that server; right? And then it's preserved there. The National Records Act does require cleartext versions of the records. You obviously - they won't let you store it encrypted.

Steve: But what we've seen is that many agencies in the U.S. government were TeleMessage clients.

Leo: Right. [Crosstalk].

Steve: It is. But they were nonetheless.

Leo: Yeah.

Steve: And I've heard from many of our listeners whose corporations were TeleMessage clients and are now looking for an alternative.

Leo: Right. Learning their lesson. Obviously the vetting they should have done of that. I mean, that diagram is clear.

Steve: So it's certainly not illegal to use Signal, yet it would be very convenient to be able to have a secure archive of Signal conversations.

Leo: Right. I think the thing is that the federal government does have a protocol for this, and the protocol wasn't followed. And really this is the real issue in general is humans. That the breaches often occur because humans don't follow correct procedure; right? And then...

Steve: Well, and this came to light because a very insecure system was in use. And the good news is we have listeners whose companies did not, you know, were not sufficiently aware that just saying, oh, sprinkle some end-to-end encryption dust on it.

Leo: Right. Yeah, TeleMessage was invented by the Israeli Defense Force. What could possible go wrong? It's just amazing.

Steve: Actually, the founder was ex-Israeli Defense.

Leo: He was - yeah, yeah, I know. It's my guess that this was always a honeypot from the Israelis.

Steve: Well, the really cool solution is just to create Signal-bots.

Leo: Yeah. Right.

Steve: And have a secure Signal-bot...

Leo: [Crosstalk] conversation. They're inside the Pentagon. They're in a secure environment. They record it. And then it's done. Job done. Instead they sent it to The Atlantic magazine editor Jeffrey Goldberg. He was their Signal-bot. You know, it's funny, he was more careful with the information than they were. That was the irony of the whole thing. Yeah, I think I like your idea. I think that's a great idea. For all we know they are implementing something similar. I mean, I would expect that.

Steve: It's open source. And there are desktop versions. You could take the desktop version and turn it into an archiver. I looked around, but I didn't find anything. Maybe our listeners will find [crosstalk]...

Leo: The NSA probably has and offers something like that, or the GSA, or somebody offers that. But, you know, you have to get people to use it. That's the problem.

Steve: Well, and it still doesn't dispel the problem of having that kind of a conversation on consumer smartphones. Because we do know that...

Leo: Phones are compromised, yeah.

Steve: Other aspects of Israeli services that we've talked about in the past.

Leo: The NSO Group, yeah.

Steve: Yup, are prying into consumer smartphones.

Leo: Yeah. Steve, as always, great stuff. And I love your clear thinking. You always come up with great solutions. They seem obvious after you say them.