Wacky Data Exfiltration

Description: This week we begin by discussing the implications of last week's LastPass breach disclosure. We look at some recent saber-rattling by the U.S.'s FTC and FCC over the disclosure of presumably private location data. We share pieces of a fascinating conversation with a Russian ransomware operator, gaining some insight into the way he conducts attacks and the way he views the world. We tell everyone about a new tracking-stripping and privacy-enforcing email forwarding service that's just come out of a yearlong beta from the DuckDuckGo people. We have another big and widespread IoT update mess to share. I have some welcome progress to report about my work on SpinRite, and some listener feedback. Finally, we're going to look at some recent goings on at the Ben-Gurion University of the Negev, which never fails to entertain.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-886.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-886-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He's got his take on the LastPass breach. I know you wanted to hear all about that. Then it's an interview with a hacker, some really interesting revelations from a ransomware hacker in Russia. Finally, he's going to talk about wacky ways to exfiltrate data from air-gapped computers. Some really interesting ideas here. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 886, recorded Tuesday, August 30th, 2022: Wacky Data Exfiltration.

It's time for Security Now!, the show where we cover the latest security news with the man, the myth, the legend: Steve Gibson. Hi, Steve.

Steve Gibson: Yo, Leo, great to be with you again for this last podcast of August. Where has the year gone?

Leo: Where has it gone, yeah.

Steve: Yeah, actually Lorrie and I were - we have an anniversary of our first date coming up here. It's five years.

Leo: What?

Steve: It's like, five years.

Leo: That's great.

Steve: Where did that time go?

Leo: Do you give - are there gifts for the first date?

Steve: Thank god, no. She's so sane. She's like, oh.

Leo: You don't have to worry about that.

Steve: No, the reason we're married is that she's completely, like, not that way.

Leo: Low maintenance.

Steve: You know, I mean, like I have to tell her that it's Valentine's Day. She says, "What? Oh." Okay. Do you care? "No." Okay, good. What's on TV?

Leo: What are you guys watching these days?

Steve: Oh, well, we are really liking "The Old Man."

Leo: Isn't that good?

Steve: Yeah.

Leo: But, well, I won't say anything. After you finish it, talk to me because it starts so well.

Steve: I totally agree. And I think I just hit the rough spot you're talking about.

Leo: Yeah.

Steve: And it's like, wait a minute. This is not what I wanted to have happen.

Leo: Bingo.

Steve: I just, yes, I loved the way it was progressing. And then, you know, then Zoe does her...

Leo: Yes, Zoe. Exactly when that happens.

Steve: And it's like, this is, like, went off the rails, unfortunately.

Leo: Round about Episode IV. It's too bad because...

Steve: Uh-huh, that's exactly right. And it was Roman numeral IV, Episode IV, in the apartment. And it's like, oh, crap.

Leo: I feel bad because I've recommended this to so many people, but I recommended it after the first three episodes.

Steve: Yes. And the reveal that we got, that surprise? Oh, goodness.

Leo: Yeah.

Steve: You know?

Leo: Yeah.

Steve: About his daughter.

Leo: Yeah, yeah, that's a good twist.

Steve: And Lithgow is still in fine form after all these...

Leo: I love John Lithgow.

Steve: Oh, my goodness. So anyway.

Leo: Great cast all around, actually.

Steve: Watching that, I think the next one is - I can't think of what the name is. It's not the timeline, it's the something.

Leo: Not "The Time Traveler's Wife." I'm sure you're not going to watch that.

Steve: No, no, no, no.

Leo: Although Lorrie might like it because the star is naked a lot, and he's very hunky. You know, it's so funny because you can see the TV...

Steve: We won't be watching that.

Leo: You can see the TV executives going, can he be naked? Yeah, he can, yeah. You know what I've been watching, and it's an old show that I discovered lately on Amazon Prime, it's called "Patriot." And I think you guys would love it. So make a note of that.

Steve: However, you were saying just briefly, and we should put it into the recording now since you hit the record button already...

Leo: Yes, yes.

Steve: You started reading - thank you. You started reading "The Singularity Trap."

Leo: I did. You were right. This is the book that last week Steve almost didn't do the show because he had, like, four pages left, and he just couldn't put it down. So I thought, well, that's pretty good. And it's by Dennis E. Hamilton, who did the incredible Bobiverse.

Steve: Dennis Taylor.

Leo: Taylor, I'm sorry. Incredible Bobiverse saga. And it is definitely that style. Same reader, Ray Porter.

Steve: Yup.

Leo: And I'm quite enjoying it. Yeah, good pick.

Steve: And it's got wit, and it's written smart, you know, it's smartly written and, yeah.

Leo: Yeah, I agree. I agree.

Steve: Okay. So we're going to talk about Wacky Data Exfiltration, brought to us by those amazing engineering students at the Ben Gurion University of the Negev, which never fails to entertain. But first we have to discuss, because boy did my Twitter DM feed...

Leo: I bet.

Steve: ...go overboard with the implications of last week's LastPass breach disclosure. We then look at some recent saber-rattling by the U.S.'s Federal Trade Commission and Federal Communications Commission over the disclosure of presumably private location data, which turns out not to be such. I want to share some pieces of a fascinating conversation with a Russian ransomware operator, which gains us some insight into the way he conducts attacks and the way he views the world, which is just a little jarring for me.

I also want to tell everyone about a new tracking-stripping and privacy-enforcing email forwarding service that's just come out of its yearlong beta from our friends at the - I wish he'd come up with a different name - DuckDuckGo. We also have another big and widespread IoT update mess to share. Then I've got a welcome progress report about my work on SpinRite and some listener feedback. And then we're going to look, as I said, at two new wacky ways of exfiltrating data from air-gapped computer systems.

Leo: Wow.

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

Leo: Wow. Exfiltrating data. Wacky ways. There have been quite a few, come to think of it, over the years we've been doing the show.

Steve: Oh, look. And I remind us at the beginning of talking about this about aiming the laser at the bag of potato chips.

Leo: Right.

Steve: Okay. So this Picture of the Week was tweeted to me.

Leo: Yes.

Steve: And it was so cool that - and I apologize for wondering if it was authentic or not. So I went back to the source, to the original tweet from the U.S. Army Chief of Cyber, who tweets from @armychiefcyber. And apparently the slogan is "Defend, Attack, Exploit."

Leo: Okay.

Steve: Yeah. And so the tweet reads: "Interested in becoming a nation-state hacker? We will develop your skills in offensive and defensive cyber operations."

Leo: Wow.

Steve: "Defend, Attack, Exploit." And then there's a link. And I've got the link in the show notes, and a link to the original tweet. Now, Leo, I have to say that I've often wondered, if I were a youngster, like what would I do? Well, even if the pay wasn't that great, the idea that you could actually, like, it would be legal for you to be attacking...

Leo: Make portable dog killers? What? Is that what - wow.

Steve: No, I mean, this is so cool. And props to them for just saying this is - look at him standing next to this big emblem there. Defend, Attack, Exploit.

Leo: It's defensive and offensive is what it is.

Steve: Sign me up, baby. Oh, goodness. Unfortunately...

Leo: Yeah. I'm not crazy about the uniform, though.

Steve: Well, and the problem is if you're really good you probably get moved into the bureaucracy. At that point I would say, okay, I'm going to take all my skills that I've just sharpened and go somewhere else.

Leo: This is very Jason Bourne, though. You really do what to do this; right?

Steve: Oh, I do. I mean, it's legal. You can attack people. Holy crap.

Leo: And you're doing it for the good guys. Yeah.

Steve: That's right. That's right. You know, we're giving the Russkies something back. It's like, oh.

Leo: The only thing I don't like, and I guess you can't expect more from the Army, is the word "cyber." Just by itself I don't like "cyber." Right? But I guess that's, you know.

Steve: Yeah, I think that's with us. We can thank William Gibson for that one.

Leo: Well, I don't mind "cyber" in conjunction with another word. But this guy is the U.S. Army Chief of Cyber.

Steve: Oh, I agree, that's a little awkward. Yes, yes.

Leo: You know why?

Steve: Cyber what?

Leo: Because they don't want to say "cyberwarfare." They don't want to say "cyberwarfare." But that's what it is; right? They just don't want to say it.

Steve: Exactly. Exactly.

Leo: You could say "cyberdefense," but then it's exploit, as well.

Steve: So we've been talking about careers in IT and in hacking. And, you know, if your particular, like, bent would suggest, I just wanted to make sure that everybody knew that this was actually happening. So very, very, very cool.

Leo: Very cool. Very nice, yeah.

Steve: Okay. So not so cool was the news of last week's LastPass breach announcement, which as I mentioned before overwhelmed my Twitter DMs. So I wanted to lead with this because so many of our listeners, myself included, are using LastPass. So I had, as a consequence, also received an email from LastPass. The current LastPass CEO, and I say "current" because it's been jumping around somewhat recently, a guy named Karim Toubba had the following to say in their online blog posting which echoed the email that he went to everyone.

He said: "I want to inform you of a development that we feel is important for us to share with our LastPass business and consumer community. Two weeks ago, we detected some unusual activity within portions of the LastPass development environment. After initiating an immediate investigation, we have seen no evidence that this incident involved any access to customer data or encrypted password vaults. We've determined that an unauthorized party gained access to portions of the LastPass development environment through a single compromised developer account and took portions of source code and some proprietary LastPass technical information.

"In response to the incident, we've deployed containment and mitigation measures, and engaged a leading cybersecurity and forensics firm. While our investigation is ongoing, we've achieved a state of containment, implemented additional enhanced security measures, and see no further evidence of unauthorized activity. Based on what we have learned and implemented, we are evaluating further mitigation techniques to strengthen our environment. We've included a brief FAQ below of what we anticipate will be the most pressing initial questions and concerns from you. We will continue to update you with the transparency you deserve. Thank you for your patience, understanding, and support."

Leo: So note that there's not a categorical denial that anything like password vaults - it's just "no evidence of."

Steve: Right.

Leo: So I feel like we're not completely out of the woods, that I'd like to know that there is in fact not merely no evidence of, but it didn't happen.

Steve: Okay. Yes.

Leo: I'm curious what you think about that. The other thing is I think this is part of the Twilio breach, that this was a follow-on on the Twilio hack, which turned out to really be problematic.

Steve: It was pretty deep, yes.

Leo: Because so many people use Twilio for authentication and other, you know, texting and so forth.

Steve: So of course we have the problem of proving a negative. So lack of evidence isn't evidence of lack and so forth.

Leo: True, right.

Steve: Okay. So the short version of the FAQ, I'm not bothering to share it all, but it was basically that they believe there is to be zero impact upon LastPass users. You know, no need to change passwords, do anything, or take any action of any kind. And I'm sure they're unhappy that this occurred since I'm sure that they hold their proprietary information in high regard and don't attackers snooping around in it.

But we've always known, since I first checked out the technology that Joe Siegrist originally designed, is that so long as the LastPass code that runs our local browser vault is not itself compromised - and that's the key, I mean, that's the golden goose there is the script in our browser that knows how to decrypt the local copy of the vault. As long as that's not compromised, the only thing we're providing to LastPass, the only thing they have of ours to lose is a very well-protected encrypted blob of entropy, one from each of their users. That's what they hold for us in the cloud which allows them to link all of our devices together. And I'm sure this is no longer unique technology. I don't know that it was back then. But though I haven't looked, I would imagine and hope that's what every other password manager also does because it's the only way to do what we all want safely.

We know that LastPass uses a strong, many-iteration PBKDF, you know, a password-based key derivation function which runs in our local browser to encrypt all of our password data before it ever leaves our local machine. So you need to have a good strong password to protect your vault. If you have that, you're as safe as you could be. And presumably, adding any of their other security measures such as multifactor authentication, hardware dongles, et cetera, only strengthens things from there.

But this leaves us with the question: With LastPass having admitted to having one of their developer accounts breached, should we change password managers? I was asked that directly by many of our listeners. And it's a worthwhile question. Lacking any additional information and no additional information is available at this point I think that's an emotional decision rather than a rational decision. Which is not to discount it. I mean, you could argue that the human race is here because of the result of emotional decisions.

Leo: Well, you could argue that Trust No One is an emotional decision, too, I guess; right?

Steve: Yes, yes. So the reason I think that, that is, that we need a rational decision, is that because there's no factual basis currently for knowing about what matters. To make an informed decision it would be necessary to deeply understand the company's policies and procedures, like as an insider, and to know exactly how this particular breach occurred. They're not saying. Their policies and procedures would tell us how they have set up the barriers, which hopefully exist, between their developer resources and their production services.

Leo: Yeah, you hate to think that it's so easy that all we have to do is social engineer one person, and then it's all gone; right?

Steve: Well, yes. And Leo, just look at what we just learned about the way Twitter operates.

Leo: Yeah, yeah.

Steve: It's like crap. Okay. But then you would also need to know that same thing about the password manager you were considering switching to. Again, an emotional decision needs no justification, whereas a rational decision is only about justification. Now, I've always been careful to draw a clear distinction between policies and mistakes. Policies are deliberate; mistakes, well, they're mistakes. When you're an employer, for example, and this is the example you and I have often used, Leo, and an employee screws up, do you fire them because they screwed up? Or do you consider that they made a mistake and have learned a valuable lesson from it? If, as a consequence of having made a mistake, they're now a better and more valuable employee, why give them to your competition?

So unfortunately we don't know enough about the inner workings of LastPass to make an informed decision about switching. Should we now be more or less afraid? How does their actual policy and behavioral security after this incident compare to the actual security available elsewhere?

Leo: And there's an interesting comparison because it's believed that the same nation-state hacker who did the Twilio attack, we know DoorDash was attacked by the same guy. They say yes.

Steve: Yup.

Leo: But Okta, Signal, and LastPass, all breached roughly the same time using similar social engineering attacks. But the one who wasn't, but was attacked, was Cloudflare. Remember this? You had this story last week, I think. They use YubiKeys. And because they use strong security, even though the social engineering attack worked, it didn't compromise them.

Steve: Yeah.

Leo: So that's the kind of thing I'd like to see from LastPass, yes.

Steve: Right. And in his note he was noncommittal. I mean, he wasn't specific. He talked about increasing their security and tightening their boundaries and things. It's like, okay. Again, so we have an example. But again, to make a change you need to know about where you're changing to, just as much as you need to know about where you're changing from. So if LastPass learned a valuable lesson, that's great. But I have no idea, and neither does anyone else. Their track record is all we really have to go on. And it's been good so far because the security architecture is good, and it's the security architecture that I'm relying upon. At the same time, as I said, presumably everybody else's security architecture is equally sound, because none of this should be rocket science anymore.

Leo: Would you recommend changing your LastPass password at this point? Would that be a reasonable response, rather than changing your password manager?

Steve: No, no, no. I don't see how that has any effect because it's the password which is used only locally to encrypt the blob which we send there.

Leo: They don't have access to that. Nor do they need it.

Steve: They never have. They don't want it. And that was Joe's original concept. So if I were starting out today, all other things being equal, I would probably choose Bitwarden.

Leo: Our sponsors, we've got to say.

Steve: Yes.

Leo: That's not why you're choosing them, I'm sure.

Steve: No. In fact, being open source, I'd be able to do the same sort of security architecture vetting that I once did with LastPass's designer, Joe Siegrist. As we all know, and as you just said and reminded us, Bitwarden is currently a sponsor of the TWiT network, and I think that's great, though it's worth noting that LastPass had never been a sponsor here at the time I chose them.

Leo: Yes. In fact, it was because you chose them I think many years later that they came to us.

Steve: I figured it was.

Leo: Yeah.

Steve: Yeah. I chose them because Joe was more open than everyone else, which allowed me to understand exactly how their system worked and why it was the proper design.

Leo: It's kind of ironic because if in fact what the bad guys got from LastPass is the source code, Bitwarden's open source. They got that already. Right?

Steve: Right. And in a properly designed system it shouldn't matter. Yes.

Leo: It's okay. It shouldn't matter. Exactly, yeah.

Steve: Yeah. So anyway, many of the flood of DMs I received last Thursday asked whether I was still using LastPass; and, if so, whether I was now planning to change. Security Now! Podcast #256 - I love that it was 2^8 - was dated July 9th, 2010, and it was titled "LastPass Security." The little summary description for it on TWiT says: "Steve thoroughly evaluates LastPass, explains why high-security passwords are necessary, and tells us how LastPass makes storing those passwords secure."

So it looks like I've been using LastPass for the past 12 years, and I still am. If they ever give me a rational reason to change, I will, in a heartbeat. And whether or not Bitwarden is still a sponsor of the TWiT network at the time, I would probably go there because openness matters. But so does inertia, and the devil you know. So anyway, I'm still using them. I don't see any reason to change. Subject to additional information coming to light, there's never been a breach that affected our stored security because of the way it's designed.

Leo: Yeah. That's what counts.

Steve: And that's really what counts. And then it's a matter of looking at the pricing and the features and what suits your model best. I just never have a problem with it. So it's not...

Leo: No, no, no reason, yeah.

Steve: It's not irritating me.

Leo: I have a very soft spot in my heart for LastPass, not only because of your support, and I used them for many, many years, but when they became the studio sponsor a few years ago they kept us on the air through COVID. If it weren't for LastPass, I don't know if we'd still be on the air. So I have a very soft spot for LastPass. I do use Bitwarden. I like the idea of open source. But I think there's pretty much feature parity between most password managers at this point.

Steve: Yeah. And really it's just inertia. It's like there's no good reason for me to leave because it works. And when there is, yeah, I'll be out of there in a hot second. But so far, so good.

Okay. What's not so far so good is that just yesterday the U.S. Federal Trade Commission - well, maybe this is good, actually - filed a lawsuit against the large data broker known as Kochava, K-O-C-H-A-V-A. Probably most of you can go to their website. I can't, but I'll tell you why in a minute. The lawsuit's complaint, that is, the Federal Trade Commission, U.S. Federal Trade Commission's lawsuit complaint alleges that the company Kochava offered for sale the precise geolocation data of hundreds of millions of mobile devices and one wonders where they got it, we'll get to that in a second - revealing potentially sensitive information in what the agency says amounted to an unfair or deceptive consumer practice.

According to the FTC's complaint, as part of its operations - these guys are in Idaho - Kochava "collects a wealth of information" about people and their mobile devices, including by purchasing it from other data brokers, and sells customized feeds.

The FTC explained that among the information it sells is precise geolocation information associated with a unique marketing ID that can be used to reveal visits to sensitive locations, such as places of worship and healthcare providers. Such data can also be relatively easily tied back to an individual by observing patterns, such as regular sleep and work locations. Samuel Levine, the Director of the FTC's Bureau of Consumer Protection, said in a press release announcing the suit: "Where consumers seek out healthcare, receive counseling, or celebrate their faith is private information and should not be sold to the highest bidder. The FTC is taking Kochava to court to protect people's privacy and halt the sale of their sensitive geolocation information."

When asked about the suit, Kochava's legal representatives did not immediately respond to a request for comment. Kochava charges clients $25,000 - and I don't know if that's monthly or what, but I saw that number - for access to its location feed, like where is so-and-so right now, I don't know - and until recently offered free samples. Kochava attempted to preempt the action by suing the FTC earlier this month, alleging overreach in proposed complaints the agency shared in July and August. Like, okay, this is the way we're getting ready to complain about you. Got any comment? And so they got sued. Shortly before suing the FTC, the company also announced a new capability called "privacy block" which it said should assuage the agency's concerns by removing "health services location data from the Kochava Collective marketplace."

So, okay. We know what's behind this; right? This is all being allowed to occur, well, first of all, the tracking is only being allowed to occur, only because it's invisible to the consumer. If tracking was apparent, it would never have grown so out of control. As we know, Apple started requiring their apps on the iOS platform to obtain consumers' explicit permission to track them outside of the app, and the result was a resounding "No!" So I'm glad that slimy companies like this Kochava are finally being put under the spotlight. It's annoying that it took the Supreme Court's overturning of their previous decision in Roe to bring this to the forefront, but better late than never.

And as I noted before, when I attempted to go to https://www.kochava.com to see what they were bragging about, Chrome told me that the domain was unknown. I got back "DNS_PROBE_FINISHED_NXDOMAIN."

Leo: What?

Steve: Of course NXDOMAIN is the error for there's no DNS listing for this.

Leo: What?

Steve: Then I smiled when I realized that was because I took your suggestion, Leo, last week and decided to experiment with NextDNS...

Leo: Oh, bravo.

Steve: ...as an advertising and tracking blocker. Obviously, those guys already know about Kochava. And they're saying, uh, no. So I'm very impressed with what I've seen so far. If I was curious, I could have quickly whitelisted Kochava.com and then gone to their site and poked around, or changed my DNS and then changed back, whatever.

Leo: You ain't missing anything. I was going to tell you. Incidentally, they offer a product to help you improve your Apple search ad performance, as well. They know a lot. They know a lot. Wow.

Steve: Yeah. So this is the world we're in. This is the data broker. And the question is, where are they buying this information? Which brings us to the Federal Communications Commission. Not to be left out, though not that it appears to matter much. The U.S.'s Federal Communications Commission has launched an investigation into mobile carriers' geolocation data practices.

Leo: Oh, yeah, sure.

Steve: Uh-huh. Last Thursday, the FCC shared responses from mobile carriers to a probe into how they handle geolocation data, and announced a new investigation into carrier compliance with the Commission's rules about disclosing how much data is stored and shared. Okay. So the FCC's Chairwoman Jessica Rosenworcel said in a press release: "Our mobile phones know a lot about us. That means carriers know who we are, who we call, and where we are at any given moment."

Leo: And where we are at any given moment.

Steve: Oh, thank goodness.

Leo: Yes. I was going to say, "Where we at? Where are we at? Where we at?"

Steve: Where we are, thank you, at any given moment. I couldn't believe that was actually going to be an official press release.

Leo: They must use Grammarly. I'm sure they fixed it up for them, yeah.

Steve: "That's why the FCC is taking steps to ensure," she says, "this data is protected." Except they're not. Anyway, good luck with that. Though I suppose this might answer the question of where slimeball Kochava obtained the information they're now aggravating, yeah, aggravating aggregating.

Leo: I like that. Aggravating, yeah.

Steve: Aggravating and reselling. Okay. So the Commission, the FCC, sent inquiries to 15 carriers, including AT&T, T-Mobile, Verizon, Google-Fi and others last month asking them to spell out their policies around geolocation data, including how long information was retained, as well as how and why, what circumstances, like how much do they pay you, it might be shared with third parties. The FCC requires mobile companies - get this - to get consumer consent for sharing information, unless such sharing is necessary to complete a service or required by law. Which is the biggest loophole ever written; right? Oh, no, we have to retain that information in order to make our cell towers work. So great, we can, we're allowed to.

Anyway, unfortunately, aside from that, has anyone ever actually succeeded in reading the fine print of the agreement that you click on with any of these companies? And what is one to do if the terms turn out to be onerous? All the other carriers use the same fine print.

Two years ago, as evidence of their lack of ability to actually do anything, in 2020 the FCC proposed more than $200 million in what they described as still-pending fines for major carriers for selling user location data without consent or appropriate safeguards. Jessica Rosenworcel, whose grammar is correct, tasked the FCC's Enforcement Bureau with the new investigation into the companies' compliance with rules requiring them "to fully disclose to consumers how they are using and sharing geolocation data." Again, a lot of good that'll do since they seem unable to collect $200 million in still-pending fines from two years ago. I would just say pull the plug and get their attention. But that'll never happen.

Justin Brookman, the head of tech policy at Consumer Reports, said that nevertheless, "The quality and specificity of answers" - that the FCC received as a result of their inquiry - "definitely ranges among the respondents; but there's some interesting, concrete information in there, especially on data retention periods." However, not surprisingly, in some cases the responses simply referred to their dense, publicly available privacy policies. You know, it's like, yeah, oh, go see the fine print. That's what everybody already checked. But some others did answer questions directly point by point.

Justin, this Consumer Reports guy, agrees with me that transparency isn't enough. He said: "People have no choice but to share very sensitive data like geolocation with mobile carriers for those products to work. There should be substantive constraints on what they do with that information and for how long they keep it." So what I wonder is, in a world where that information can be sold to third parties in real time, that is, it's not like it's being typed up or printed out and emailed in boxes somewhere, that information can flow out the moment it's captured. And where its timeliness makes it valuable, it's unclear to me whether "retention time" anymore matters at all. That sounds like, you know, pre-communications is free sort of timescale.

Harold Feld, who's the Vice President with a company called Public Knowledge, also called for regulatory action, saying the FCC should "set new rules of the road" for mobile carriers' privacy. He said: "These letters show that, despite the constant invocation of 'industry standards' and 'best practices,' carrier geolocation data practices are all over the map." For example, the length of time carriers retained location data, as determined by proximity to cell towers, ranged widely, and as long as five years in the case of AT&T.

So my feeling is, unless we make them delete it, they're not going to. But again, if they are allowed to sell it immediately to third parties, retention time no longer matters at all. They claim that they must keep it for business purposes and to maintain the health of their networks. Fine. Simply outlaw its sales to any other entity, period. But that's not

going to happen. The FCC appears to be toothless to me. They completely ceded control over broadband privacy during Trump's administration. And while the FCC still theoretically has substantial regulatory authority over mobile phone carriers, the carriers appear to simply be ignoring the FCC. So I don't know. It would be nice to have something happen at the federal level. Maybe it's going to be at the state level.

Which brings me to California. In very late-breaking news, only a couple hours ago, Techdirt's Mike Masnick reported...

Leo: Ooh, is he mad. Ooh, baby is he mad.

Steve: Yes.

Leo: And I don't blame him.

Steve: Yes, he is. The California State Senate just passed what he described as three horrific new Internet regulation laws, apparently written by bureaucrats who have no idea how the Internet works. Since this just happened, I haven't had any chance to look into it. But if it's interesting, we'll talk about it next week.

Leo: Oh, it is, and we'll be talking about...

Steve: Do you know more, Leo?

Leo: Yeah, I know all about it. We'll be talking about it tomorrow on TWiG. It is one of those things where it sounds on the surface to be very good. New York Times published an article about it, you know, essentially saying, yeah, it's good to have privacy. It doesn't understand how the Internet works. It's not going to solve the problems it's intended to solve. And in fact it's probably going to make them worse. And it's certainly going to be an onerous burden for us because any, well, one of the laws is to protect children, people under 18. But the COPPA, the Child Online Privacy and Prevention Act, protects kids under 13, but against sites that are aimed at kids. This affects all sites. So if you think an 18-year-old might visit your site - well, guess what, they do, they do - then you have to do a number of things to protect them, including age verification.

Steve: What?

Leo: You have to make sure that nobody is under 18 who's visiting your site; or, if they are, that you mitigate any hazards to them. Well, Mike's point is, well, you're asking us to figure out how people are when they visit the site?

Steve: Every single visitor.

Leo: Every single - and collect that information. That's not exactly privacy forward.

Steve: No.

Leo: Nor does it protect anybody. So this is - I think your summation is exactly write, written by bureaucrats who have no idea how the Internet works. It's politically probably very popular because it looks good. Looks like it protects children and protects privacy. Does not. Anyway, we'll talk about this tomorrow.

Steve: And 100% of the Senate voted for that.

Leo: 33 to nothing.

Steve: Yeah.

Leo: Nobody voted against it in California. Now, the last buttress against this will be the governor. And what Mike's hoping, I think a lot of people are hoping, is that Governor Newsom will hear from people like Mike Masnick and say, oh, yeah.

Steve: And maybe that he's tech savvy enough to understand what this means.

Leo: Right. What people like The New York Times say is, well, the giants don't like it because of course they want to collect more information about us. But imagine. Thing is, Facebook already knows how old you are. Google can easily figure that out. But Security Now!, GRC.com? TWiT.tv? Do we want to start collecting age information about everybody who visits?

Steve: No. And in fact when I set up GRC's forums, I explicitly removed that from the signup sheet and from any criteria because I didn't want to ask for it. I don't want to know. I don't care.

Leo: The legislature's response to Mike is, well, you know, the AG, the California Attorney General, gets to decide who's prosecuted. He's not going to look at TWiT and say, oh, yeah, this is a hazard to 18 year olds. But Mike's point is, oh, great, so now you give the AG a tool that if he doesn't like somebody, he can attack them. That seems like...

Steve: On the basis that you're not collecting visitor age information.

Leo: Right, right.

Steve: Yeah, we'll just add that to the cookie banner, Leo. We'll just add another field.

Leo: Oh yeah, just like that.

Steve: You know, what year were you born?

Leo: By the way, all of a sudden, if you start collecting age information, now a whole range of GDPR regulations apply to you because you are collecting personally identifiable information. Right? So it opens you up to this whole can of worms. Anyway, don't get me started. I agree with Mike.

Steve: Lorrie's been talking about New Zealand of late.

Leo: Yeah, sounds better and better. And by the way, no Internet presence. Just stay off the Internet. It's a bad idea.

Steve: Back when I was writing the TechTalk column for InfoWorld, I, as all columnists, had a copy editor.

Leo: Yes.

Steve: Mine was a great guy named Michael Miller.

Leo: Oh, I know Michael, yeah.

Steve: Yup. And he once said to me something that, it was like, what, 30-some years ago? And it's just stuck with me ever since. He said, "Well, Steve, you know, mostly I just go through your columns searching for the word 'which,' and I change them to 'that.' Because you do that." And I go, oh. And so of course I've been self-conscious about it ever since. I like the word "which." And sometimes it seems better to me than "that."

Leo: You can overuse "which." I don't know if it's the same Michael Miller, but I think it is. He became Editor in Chief of PC Magazine. We were talking about him this morning, as a matter of fact.

Steve: Yup, that's Michael Miller. That's my Michael Miller.

Leo: Great guy. Really like him. And he's done very well since. I don't think it says in here "Former copy editor for Steve Gibson."

Steve: Oh, he'll know. Eight years he had to go through with his - thank god he had copy and replace, yeah, find and replace.

Leo: He was a great guy. Or he is a great guy. I really like Michael, yeah.

Steve: Yup. So the guys over at the publication The Record had a lengthy conversation with a Russian ransomware attacker by the name of Mikhail Matveev.

Leo: Interesting.

Steve: And although I didn't think that much of the conversation, which revolved a lot of the squabblings among adversarial ransomware groups, would be that much interesting to our listeners, Mikhail's answers to a couple of the questions were interesting. So I've selected a few bits out of that longer conversation to share. And I should mention that this is a translation from Russian, because the conversation was held in Russian, so the semantics will be a bit non-English. And I've also edited it a bit since this young man's choice of descriptive language was a little bit blue. So it was not safe for work.

Leo: Well, he is a hacker. I mean, come on.

Steve: Yes, he is. I didn't have a problem with it, but we've got a large listening audience.

Leo: Yes, no, I appreciate that. We don't want to have to scan your face to find out how old you are.

Steve: No. Dimitry from The Record asks: "How often do people from different affiliate programs compete in the same network to extort victims? Have you had such situations?" In other words, Dimitry was asking, are there ever collisions among different attackers? And Mikhail says: "This happens often." What? "Especially when several people own the exploit, or pour logs from the same traffic market if we are talking about extracting initial access credentials with a stealer."

He says: "I took some source codes, so-called 'proof of concept,' from GitHub and modified them. If you remember, there was a well-known CVE for the Fortinet VPN. We found it with one programmer from the forum. Based on the list of IP addresses, we got approximately 48,000 entry points. I was very surprised then, really shocked. But we did not even work through 3% of this list. Not enough time.

"And when others well, let's say our competitors began to use this vulnerability, there were intersections across networks. I often went into a network already locked" - and by the way, when he says "locked," that's his term for encrypted. So he went in, and everything was, all of the servers were already encrypted. So he says: "I often went into a network already locked by someone and didn't touch them because it's not my job to encrypt for the second time. But some guys over-locked networks. They come in and see that it is encrypted. And so that nobody gets it, they encrypt it again. There were cases where the guys and I just crossed paths on the network during development" - that is, development of their presence in the network - "exchanged contacts, and somehow discussed what to do next. We basically always agreed.

"And it even happened that we then jointly did some other projects. In the summer of 2022" - that is, this summer, he says - "this happens all the time because everyone is hungry for the material. How can we get to the initial access? Actually, there aren't many options. There are vulnerabilities, such as RCE [Remote Code Execution] in various products of VPN devices, everything that can give access to the network. Or a network access login from stealers. But basically, everyone is now flooded from traffic exchanges, and there is little unique traffic. And those who have it, they pour just for themselves or are already working in some teams, so it's absolutely normal that there is a conflict of interest on the networks, and now it will be even more."

Okay. So I thought Mikhail's comments that they were only able to exploit 3% of the list of 48,000 Fortinet VPNs because there's not enough time. In other words, he's saying there really is an active race when a new patch drops and a proof of concept is made available for something like a critical remote access vulnerability. So these cretins are actively watching everything, waiting for the first glimmer of a newly discovered problem. And they realize that there is going to be a lot of the systems patched quickly, so they're not wasting any time.

And significantly, they are not finding any of these problems themselves. These are not high-end security researchers gone bad. They're living off of the interval in the delay to patch. They're not good enough to find the trouble themselves. But they are good enough to quickly weaponize a working proof of concept when it's posted to GitHub, then immediately turn around and employ it to gain entry wherever they can. And basically what he's saying is there's now a lot of them, all basically competing for access to opportunities that appear whenever they do, and so it's who can get in there first.

So Dimitry asks: "Tell me about some attacks that stood out to you. Which was the fastest? How long did it take from the first penetration into the network to receiving the payment?" So Mikhail says: "There were many interesting ones. But I would like to sum it up, before talking about the attacks. There are small networks, there are medium networks, and there are very large networks. And I'll tell you, it's much easier to work with a network of an organization with $1 billion of revenue than in a network of an organization that has income of $9 million.

"I'll tell you why. There are many more computers that are easier to hide on and easier to navigate than in a small network where you are limited. You have to move very fast. And when I started my career" - I love the word "career." It's like, this is a career. Okay. "When I started my career, I started with BlueKeep a vulnerability in Microsoft Remote Desktop. I hacked five small networks per day because I had to go in and do it right away. But, as I progressed, the time I spent on the hacks increased."

He says: "My longest development, probably everyone has heard about the Capcom company. I got there through a Fortinet vulnerability. As a matter of fact, when I went there, I was a little surprised that everything was in Japanese. There is no hierarchy, there's no division into departments, and they have everything in a big heap. I found a dead domain admin. That is how the name Babuk" - which is one of his monikers - "appeared." He said: "Capcom had an admin Babak, or Bambook. And when I found this administrator, I realized that no one uses him, but he was an enterprise type." Okay, so Mikhail is explaining that he found an abandoned active administrative account which he was able to use. And he took "Babuk" as one of his several aliases from them on.

He said: "The fastest attack in my life happened as soon as I got the ProxyLogon vulnerability. At that time, I had a programmer on a grant who was finalizing the exploit. One of the interesting networks was a logistics company in the Netherlands. Large warehouse. Very large warehouse. I got in and immediately obtained the domain admin tokens. These guys weren't very security conscious and didn't worry about anything. I remember I went there at 8:00 p.m. Moscow time, and at about 4:00 a.m. Moscow time" - so that was eight hours later, he says - "it was already all locked up." Meaning he'd encrypted the works. He said: "From 6 a.m., the administrator wrote to us in a panic, to which I told him, 'Bro, wait for the supervisor.'"

So anyway, he's saying that somebody realized something was wrong, and Mikhail didn't want to talk to an underling. So he said: "Looking around the network, everything seems to be simple and clear. They have an administrator's domain for us. The password was the same for everything."

Leo: Oh, boy.

Steve: Uh-huh, "On hypervisors, on a backup server, in the work group, everything. After analyzing the network, I found a WIM Windows backup system. I could get all the passwords from it." Right? Because it's an offline backup. He says: "And thereby got all the backups, although their backups were so bad." He said: "They just backed up to the NAS." He says: "I went to the NAS and formatted it. Went to ESXi, encrypted, and then after about an hour he wrote to us."

So he says: "The admin wrote right at midnight. He said, 'I would like to resolve the issue.' I said that the issue could not be resolved because he was not a boss. In the morning" - this is Mikhail. "In the morning I had to fly to another city. I remember sitting at the airport. The company writes to me: '$2 million. Transferring $2 million.'" He said: "I have never had such an amount in my wallet. I get on the plane, realizing that I have a laptop with $2 million." He says: "Well, I gave them decryptors, and when I arrived, I opened the chat."

He says: "Damn it, something is not right there. They just yell 'You destroyed VMDK.' He says, of course, that's the file format, right, for ESXI virtual machines. So they're screaming at him: "You destroyed VMDK." He said: "I tried to figure it out and asked for VMDK samples. But the VMDK files are 0KB."

Leo: Oh, that's not good.

Steve: "So," he writes, "everything is screwed. I am writing to this developer who created the exploit for me, 'How could this happen?' He says, 'Well, I don't know,' he said, 'something broke.'"

Leo: Nice. These are quality people we're talking here, I'll tell you.

Steve: And they asked, he says: "And they asked to return the money. Well," he said, "we had no choice but to block them."

Leo: Great.

Steve: "So," he said, "we scammed them for this money. I still blame myself for this. It was the fastest and most solvent attack I've ever done."

Leo: Besides the fact this guy is an absolute scum bucket, the thing that really strikes me is how invulnerable he feels. He's confessing to at least two major crimes, Capcom and this Dutch warehouse, with absolute impunity. The Russians don't care.

Steve: Yeah. And he says something in a minute that I just - okay. So Dimitry says: "How do you see the ransomware industry in three years? Will ransomware remain the best monetization model for cybercriminals, or will they move on to something else?" Mikhail says: "It's like how carding used to be popular, and there was a lot of money in it, but now it's dead. And ransomware will soon die, not in three years," he says, "but sooner." He says, and I disagree with that, but we'll see. He says: "Literally everything" -

now, this is interesting. "Literally everything has changed over the last six months." And remember where he is. He's in Russia. He says: "Since the beginning of the special operation in Ukraine, almost everyone has refused to pay."

Leo: Oh, good. Good.

Steve: He says: "I often encountered people who wrote to me in the chat, 'You are a Russian occupier. Be content with $10K, and we won't give you more. At least take that.'" So he says: "Return on investment has completely fallen in the last six months."

Leo: Aww. Aww.

Steve: Oh, boo-hoo.

Leo: What is this? Geez.

Steve: I know. "Return on investment has completely fallen in the last six months." He says: "It became difficult to work in general." Poor baby. He says: "If it dies" - meaning ransomware - "it dies. You need to come up with something new. But ransomware is worse than heroin. I haven't tried heroin, but I've seen people who are on it, and I'll tell you this: Ransomware is worse than drug addiction. There is no such money anywhere as there is in ransomware."

Leo: [Sigh]

Steve: I know. "I even compared it to drug dealers from Hydra, the world's largest dark net marketplace, which was shut down this year. They earn less than we do." Okay. So he's calling this "earning money." He finishes: "But at the moment, ransomware remains the leader in monetization. There are no other schemes on the Internet that would carry more monetization, or I don't know about them yet."

Leo: Mm-hmm.

Steve: So as I read that, I'm struck by how casual Mikhail is about being a criminal. There's an utter lack of morality. He did appear to feel badly that his decryptor didn't reverse the encryption of the large warehouse's VMDK files. So he got $2 million without returning their data.

Leo: Aw, yeah. He feels bad.

Steve: Yeah. Well, and he had to block them because they were screaming at him.

Leo: Yeah, hmm.

Steve: So but what seems to be utterly absent is the idea that extortion itself is wrong.

Leo: Right.

Steve: He talks about it as a "career." Like it's a legitimate profession that he's in. Like his parents would be proud. As though, if you have leverage over someone, using that leverage for your own personal gain at their loss is acceptable.

Leo: Horrible.

Steve: So anyway, I thought that everyone would find this interesting. These guys are not geniuses. They're computer savvy. They use other people's tools to force those abroad meaning as not in Russia to give them money.

Leo: Yeah. They're like criminals anywhere. They have zero moral compass and I'm sure justify it in their mind. And, oh, it's horrible.

Steve: Yeah. So, and I just don't cover the continuing ransomware problems. But, I mean, because I know our listeners are like, yeah, yeah, yeah. But, I mean, I skip over story after story after story.

Leo: Oh, yeah. Same with breaches. I don't even bother talking about breaches anymore. It's nonstop.

Steve: Yeah. Yeah. So a bit of good news. The privacy-centric DuckDuckGo has had what it calls an "Email Protection" service in beta since July of last year, so more than a year. But they've just opened it to the public. It looks like a very useful and completely free service. So our listeners might want to jump over and grab their name or their favorite handle quickly before it's taken. So I'll explain how to do that first. Then I'll tell you why this seems like a nifty service.

To register, go in a web browser to duckduckgo.com/email. If you don't, as you probably won't, currently have their browser extension installed, you'll need to do that first. You can remove it later since it's not required to use the service once it's set up. Although you do need it to manage the service, and there are some cool management stuff I'll explain in a second. So you'll be asked to provide a username which has not yet been taken. And it will become sort of your base or default @duck.com email address. So whatever name or handle you choose @duck.com will be your default email address. You also provide an email address which will receive cleaned and formatted and forwarded email.

Okay. Oh, and so after you install the browser extension, go back to DuckDuckGo.com/email. Now it'll say, ah, and then you'll be able to set up your account.

So what does this all get you? Their Email Protection is DuckDuckGo's dedicated email forwarding system which strips advertising and profiling trackers from email links, scripts, images, media, all that crap before forwarding them to your registered forwarding email. When you receive the forwarded email you'll also see a short report which has been added to it of how many trackers were removed, which companies were

responsible for their injection into your email, and more. DuckDuckGo says that after a year of running the beta program, 85% of all emails on their beta testers' communications, contained trackers of one form or another.

So at that point anything sent to "yourusername@duck.com" will be forwarded after being cleaned and reformatted. And in a very cool feature, Email Protection also provides users with unlimited disposable and dynamically manageable private addresses to use on sites you want to supply with a per-site or not your primary email address. These can later be deactivated if spamming to that address becomes a problem. You can ask for as many of these throwaway email addresses as you need. And of course using unique email addresses confers some of the benefits of using unique passwords on sites. In the event of a website's data breach, the linkability of your identity to any other of your identities online will be dramatically reduced.

So messages passing through DuckDuckGo are never stored by DuckDuckGo. They make that very clear, while what small amount of accounting and forwarding information is kept for operational reasons is deleted within 30 days after the account's closure. So if you close the account, within a month it's gone. No long-term footprint. And even though email is forwarded to your real email address, it's still possible to reply to those emails, which will then come from a Duck.com address. So this can be useful where anonymity would be important.

So it's 100% free. It has a user-friendly dashboard for quickly configuring forwarding addresses and making on-the-fly changes. You can manage account settings and all that. In addition to a browser extension, there are apps for Android and iOS which allow for the same sort of dashboard management in those apps.

So, okay, I still don't like the name, but the service seems pretty cool. Since I run my own server at GRC.com I'm able to, and I do, create tons of email addresses, I mean email aliases, exactly for this purpose. So I'm giving somebody I'm not sure about an alias. And if I ever start getting spam there, well, first of all, I know where it came from, and I'm able to terminate the alias. So this is that sort of a service which in addition to that is filtering email to remove tracking crap from it. So anyway, seems like a neat deal. We know DuckDuckGo and that they really are privacy centric. So I wanted to make sure everybody knew that this new service had just come out of beta.

Okay. Another big IoT mess. And each one of these that we talk about is a lesson. I'm not going to get preachy because I know that can get tiresome. But the state of the IoT industry brings my blood close to a boil every time. Here's what's going on this time. The cybersecurity firm CYFIRMA, C-Y-F-I-R-M-A, recently published a report describing a long and still outstanding security threat created by insecure - and these are like commercial grade - Internet surveillance cameras produced by a U.S.-based firm Hikvision, H-I-K-V-I-S-I-O-N. They're located down here near me in Southern California, I think City of Industry. Anyway, one year ago, a year ago in September 2021, in response to a discovery by security researchers which was given CVE-2021-36260, Hikvision did the responsible thing: They published a firmware update to correct a serious vulnerability.

Unfortunately, it's for a camera. Okay. The researchers discovered that the Hikvision cameras were vulnerable to a critical command injection flaw that's easily exploitable via specially crafted messages sent to the camera's vulnerable web server, which is what it exposes to the Internet.

Okay. So that was then. Today, CYFIRMA analyzed a sample of 285,000 Internet-facing Hikvision camera web servers. They found that today, a year later, roughly 80,000 are still vulnerable to exploitation. And this, of course, is the problem. Some contractor you've hired purchases a camera, or 50, and installs them. They set them up, send you an invoice, and move on to their next job. Meanwhile, you have some number of web

servers on your network which can be taken over remotely. And the takeover is not theoretical.

There have been two known public exploits for this CVE-2021-36260. One was published in October of 2021, and the second in February of 2022. So that's known public exploits published, meaning this is exactly what Mikhail is sitting around in Russia waiting to see pop up on the Internet, and he jumps on it in competition with all of the others of his ilk. In December 2021, a Mirai-based botnet called "Moobot" used one of those two publicly published exploits to spread aggressively and enlist those cameras into DDoS swarms. What's generating those record-breaking DDoS attacks which now force everyone who needs to remain online to move behind and pay for DDoS protection? Exactly these kinds of IoT devices. It may include thousands of compromised Hikvision cameras. There are 80,000 available today.

In January of this year, CISA alerted that that CVE-2021-36260 was among the actively exploited bugs in its list. CISA warned organizations that attackers could take control of devices and that they should be patched immediately. Yeah, they should have been patched last September. How'd that work out? As I noted, those 80,000 still-vulnerable surveillance cameras were just recently enumerated. The cameras are very popular, and they appear to be industrial grade, as I said. Hikvision has an impressive-looking website. In CYFIRMA's report they tracked those 80,000 vulnerable IPs back to 2,300 organizations across 100 countries. None have applied the security update which is now nearly a year old.

CYFIRMA's report notes that Russian-speaking hacking forums often sell network entrance points relying on exploitable Hikvision cameras that can be used either for botnetting or lateral movement to gain entrance into the organizations where they're deployed. Like I said, just what Mikhail is looking for. I have a chart in the show notes showing the geographic breakdown of the cameras' locations. Most of them are located in China.

Leo: Because it's a Chinese company. So not surprising, yeah.

Steve: Yes. There are 12,690 of them. And the United States has the second most, 10,611. While Vietnam, the U.K., Ukraine, Thailand, South Africa, France, the Netherlands, and Romania all count between 7,000 and 2,000 vulnerable endpoints each. So, wow, lots of vulnerabilities in those companies.

Leo: You might be interested to know that it is controlled by the Chinese government. It is not, I mean, it's a private company, but the majority of shares are controlled by the CCP. And it is used in many police surveillance systems all over the world.

Steve: I was wondering about that, too, because we don't even talk about if you compromise that, obviously you can see whatever those cameras are seeing.

Leo: That's the least of it, though; right?

Steve: Yeah.

Leo: Yeah. Leading the future of AIOT, they proclaim.

Steve: I know. Yeah. So the credential lists for those cameras pop up in hacking forums often. And I still doubt that the public at large understands the danger that's represented by the casual attachment of high-tech devices to their network. As long as that remains true, purchasers won't know that they need to consider the operational life cycle of such devices. We've all been trained now about OS and Smartphone updates. But there's just no awareness that your thermostat needs to be updated, if some third party is not taking care of that for you. So we really are in the Wild West of Internet IoT.

Okay. Over the past weekend I posted two status updates to the grc.spinrite.dev newsgroup. The first posting had the subject "Friday Night Update." And I wrote: "Gang, I just finished the complete read-through of SpinRite's DynaStat system." It's like the core of its data recovery stuff. I wrote: "I've been slogging my way through it for the past week or so. It's extremely involved, and it was working once. I wanted to be certain that I hadn't done anything to break it with all of the changes I've made the I/O driver abstraction and the relocation of several working buffers into high memory. Since they did affect the DynaStat code deeply, I've had to work my way through every code path. It's still going to need extensive testing, but that will be joyful since it will mean that SpinRite is essentially working and just needs to have the final bits of debris eliminated.

"With this done, I now need to finish the comparatively trivial task of updating the rest of SpinRite's main processing loop, the data inversion media testing, et cetera. And then it will be ready for the thorough testing of all of its main data recovery loop. But we're definitely getting tantalizingly close."

So that was Friday night. Then "Saturday Night Update." I wrote: "Okay, I'm done. This is not to say that I have any" - I have mail. Thank you very much. "This is not to say," I wrote, "that I have illusions that it could possibly run yet. There's no possibility. But I have finished working through all of the code, and now it'll be up to SpinRite to show me where it's not yet ready for primetime.

"What I plan to do next is to get it actually running so that it would appear to the casual observer to be working. That'll still be a chunk of work since I've deliberately not allowed it to begin execution. It's certain to explode fabulously. But before long, it won't be exploding anymore when it runs. At that point, when there's no longer anything obviously wrong, I'll verify that it's actually doing something useful and that all of the various data recovery paths, several of them new, are working as they're designed to. And then it'll be done."

So I just wanted to share, with everyone here who is not following along with the blow-by-blow in the SpinRite development group, where things stand. Tonight, after the podcast, I will begin running SpinRite and fixing everything that doesn't run, since as far as I know it all should. Once everything appears to be running, I'll then begin the work of carefully inducing various sorts of media read-and-write failures and carefully watch SpinRite deal with each type of problem to make sure it's doing the right thing.

And by amazing coincidence, a listener of ours, Ameel Khan, sent me a Twitter DM which I saw this morning. He said: "Hi Steve. Love the show. Been a regular listener for 16 years now. Check out this video of John Carmack talking about the importance of using a debugger while you code." So the YouTube video that Ameel linked to is an interesting 15-minute conversation with the of course legendary coder John Carmack. I have the link in the show notes and it's our GRC shortcut of the week, so grc.sc/886. That'll bounce you over to the YouTube video.

What I learned by watching the video, to my surprise, is that John and I code in exactly the same way, that is, with exactly the same philosophy. Our listeners will remember that at the beginning of my return to working on SpinRite, everyone heard me talking about setting up a comfortable and smooth debugging environment before I did anything else. And you've heard me mention it over and over since then. My wife Lorrie lived with me grumbling about that for several months while I struggled to get everything working exactly the way I wanted. In my case it was challenging because my target environment for the debugging was MS-DOS; and to do the sort of debugging I wanted to do, I needed a real-time link between a state-of-the-art 64-bit Windows machine and a 16-bit real mode DOS machine. And that has become much more tricky as the years have separated these two worlds.

Anyway, I thought it was interesting that John's code writing philosophy and mine are the same. Rather than trying to guess what's going on, rather than attempting to debug in our heads, we both immediately go to the debugger to watch the code execute step by step. As I've often noted here, something about the programmer's ego prevents us from seeing what the code actually does. We see what we want it and expect it to do, right up until the debugger slams our face into the reality. At one point John notes that tools that are easy to use get used, whereas tools that are difficult or cumbersome tend to only be used as a last resort. He and I have apparently both learned the lesson that having a comfortable and easy-to-use debugging environment is the way to get the best possible code written.

So, thank you, Ameel, for sharing the link to that conversation. And for any of our listeners who are interested, grc.sc/886.

Okay. Oh, and a couple more little bits of closing the loop. Vlad Jirasek tweeted: "Hi Steve, I have an update on this." Actually it was on an update from last week. He said: "I pressed Cybereason to clarify whether the escalation of privileges would have been successful if the users were not part of the local administrator group, and they confirmed it." And then he sent me a LinkedIn link to their dialogue. And he said: "Might be good to mention on next Security Now!. Even Microsoft is saying that removing admin privileges makes over 90% of attacks ineffective."

Anyway, so what this guy sent to Cybereason, who we were talking about last week, he said: "Very nice report, thank you. However, may I ask why you do not mention recommendation for computer users not to be assigned administrator privileges as one of the key controls protecting them against the escalation of the attack? If an attacker is not a member of local administrator group, then running fodhelper.exe" - which we talked about when we did this whole walkthrough of the analysis of the attack - "will not give attackers the administrative privileges by bypassing UAC. Am I correct?" he asked.

Cybereason, to their credit, replied. They said: "Thank you. We should have previously addressed that the point of the article is not to be exhaustive in terms of recommendations. In the case involving Bumblebee, users were already in administrator group, and UAC bypass worked. But you are correct, users need to be in administrator group. The article is focusing on post-exploitation. The recommendations list is not exhaustive."

So I thought that was interesting. Remember that we've talked about the way Microsoft has basically compromised the whole problem of running as a non-privileged user, but not making it burdensome to get root or admin privileges. In a traditional Unix or Linux environment, you are typically running not as the root user. You can do lots of things. But there are certainly low-level admin things where you need to logoff as the user or upgrade your rights using Linux and Unix commands to the root privilege in order to get something done. And you're able to run a single program under those privileges.

The way Windows does it is they developed this notion of a split token where you actually have two different security tokens. You're running with one with lesser privileges usually. But that's what the UAC does, the User Access Control, is to switch you dynamically to the admin privilege token. So his point is, and it's really worth remembering and highlighting, it's not - it would be interesting to try running Windows without the ability to elevate. I know you can, but the typical end user would probably find it more annoying than was worth the trouble or worth the added security. Maybe not so the enterprise user, where they're not supposed to be making deep changes to their system. So removing the ability to elevate by taking them out of the admin group, that's worth remembering as a possible way of mitigating, as Microsoft has said, nine out of 10 of the attacks, which do require admin elevation.

And finally, Ed McKiver, whose Twitter handle is @OhWellDamn2010, I don't know why. He said: "Hi, Steve. FYI, I canceled my LastPass Premium subscription today," he said, "(due to the recent close-call security breach)." He said: "I've had LastPass since they were a sponsor on TWIT, you gave your thumbs-up to their software/encryption, and before LogMeIn purchased the company. I'm trying to limit my exposure with my password managers" - plural, and we'll see why in a minute - "now to just one. I've used Passwords Plus from DataViz since v1.0 when it was sold..."

Leo: Oh, my god.

Steve: "...on a 5" floppy disk."

Leo: It used ROT13 for password protection, I believe.

Steve: Did we have passwords back then?

Leo: There was only five.

Steve: Leo, that's when you went, well, we know that yours was...

Leo: DataViz.

Steve: We know that yours was monkey.

Leo: Yeah.

Steve: When DataViz was v1.0.

Leo: Wow.

Steve: I didn't know we had passwords, but I guess we did. Anyway, he said: "They recently stopped all support for their product" - okay - "and their CEO decided not to move over to a subscription option in order to keep it profitable."

Leo: Good, good.

Steve: So people stopped using DataViz, I guess. And really, where are you going to stick a 5" floppy these days? That's going to be a problem.

Leo: I can give you some ideas, but okay.

Steve: Yeah. You've got to roll it up first.

Leo: Yeah.

Steve: Anyway, he said: "I tried mSecure Premium as the recommended password manager to replace Passwords Plus, but decided to cancel that password manager today, too." I guess he was in the mood to cancel these managers. He says: "As I found their tech support severely lacking." Oh, yeah, you don't want to ever talk to anyone's tech support. He said: "It seemed to me that mSecure was a one- or two-man operation." Okay, I'm kind of, well, we've got Greg, and we've got Sue. So I guess that's three people.

Leo: Two and a half.

Steve: Yeah. He says: "I'm sticking with Bitwarden as they have the best options, prices, and they also support the YubiKey. Thank you as always for your great work on Security Now!. I'm looking forward to SpinRite 6.1 since I've been a subscriber since v1.0." And yes, it also had a 5" floppy disk in the beginning. Thank you, Ed, and congratulations on no longer using 20 different password managers, whittling it down to just one.

Leo: Wow, yeah.

Steve: Yeah. So, okay.

Leo: If you want a break, why don't you take a little break before we launch into this; yeah? You've been talking for a long time. And I will take over for a minute. I can feel the tension building in your throat.

I just want to do a plug for Club TWiT. This is how we're kind of smoothing out the ups and downs in advertising. This was an idea that came to Lisa during the pandemic, and it's really been a boon to us. What do you get in Club TWiT? You get, for $7 a month, that's all, ad-free versions of all our shows, this show plus everything else we do. You get access to a really fun social media site, I think, a Discord. Discord is where you can chat about the shows, but also about anything else on your mind, every geek subject under the sun. We even have our own Minecraft servers. We have a trivia contest going on in there.

Plus we have some Discord-only shows, which you will also get access to, either live or with the TWiT Plus feed, which is a separate feed just for TWiT Club members. We

do a lot of stuff in the Discord, a number of shows. For instance, Hands-On Mac, Mikah Sargent's show is for club members only right now. As it grows and we get advertisers, it will certainly become public eventually. But same with Hands-On Windows. No? Never? Okay. Never, Lisa says. We also - so subscribe; right? That's the idea. We want to make it desirable. We also have the Untitled Linux Show. We do a lot of events. We just did Stacey's Book Club. We're going to do that again. Our community manager Ant Pruitt is planning more events in the future. So there's a lot of reasons to join, and for 7 bucks a month there's hardly any reason not to join.

Now, I should mention, if you just want Security Now!, I know some of you do, you can buy that from the iTunes Apple podcast for 2.99 a month. That'll give you the ad-free version. I think for a few bucks more getting the whole thing is a great thing. And it really helps us out. So go to TWiT.tv/clubtwit. There's a yearly plan, as well. There's enterprise plans. Check it out. And that's where we also have information about buying individual shows, including you can buy the Hands-On Mac or Hands-On Windows show. Club TWiT is at TWiT.tv/clubtwit.

And now, return to Steve Gibson and our topic.

Steve: I just looked up Michael Miller on LinkedIn. And not surprisingly, we have a couple of mutual connections, Scott Mace and Evan Katz.

Leo: There you go. And by the way, Michael Miller still works for Ziff Davis, of all things. He works for their investment arm as their CISO. So he's still in the biz. But great guy.

Steve: Cool.

Leo: All right. Now time to tell us what Wacky Data Exfiltration is.

Steve: Oh, and thank you for that pause to refresh.

Leo: I thought, you know, you've been going a long time. I thought I should.

Steve: I did need it. Okay. So through the years we've had fun considering all the various ways that Dr. Mordechai Guri and his student researchers at Israel's Cyber Security Research Center of the Ben-Gurion University of the Negev, which I also love saying every time I can, have come up with for secreting information from air-gapped computer equipment. That's like a hobby of theirs.

So we'll all recall picking up the vibrations from the surface of a bag of potato chips sitting unnoticed in a conference room. There was also, in a party setting, the balloons were known to be vibrating to the conversations being held around them. And remember there was a plant whose leaf was vibrating. And so, yeah, all that. Anyway, there have been many of these such inventions, all of which they developed and actually pulled off in order to determine the feasibility and the achievable information transmission rate. So in the past week we have their reports of two additional covert information leakage channels.

The first is actually one that we've discussed in the past. That's the blinking LEDs on network interface cards. Now, I was quick to discount that since the LEDs, as anyone knows who's actually looked at them, don't actually blink in time with the data. Although there's no hard and fast standard for the way they do blink, I notice on my equipment they're blinking for the same data in different ways, you know, rates and just different styles, in general they just show a flash when there's data activity on the line in either direction, and the light is on enough for you to be able to see it. But knowing that did not deter these intrepid Israeli researchers.

In their paper entitled "ETHERLED" - E-T-H-E-R-L-E-D. It's titled "ETHERLED: Sending Covert Morse Signals from Air-Gapped Devices via Network Card (NIC) LEDs." And they explain: "Highly secure devices are often isolated from the Internet or other public networks due to the confidential information they process. This level of isolation is referred to as an 'air-gap.' In this paper, we present a new technique named ETHERLED, allowing attackers to leak data from air-gapped networked devices such as PCs, printers, network cameras, embedded controllers, and servers. Networked devices have an integrated network interface controller (NIC) that includes status and activity indicator LEDs.

"We show that malware installed on the device can control the status LEDs by blinking and alternating colors, using documented methods or undocumented firmware commands. Information can be encoded via simple encoding such as Morse code and modulated over these optical signals." I wouldn't use Morse code. I'd use the encoding used hard disk drives because that's serial also. But anyway. "An attacker can intercept and decode these signals from tens to hundreds of meters away. We show an evaluation and discuss defensive and preventative countermeasures for this exfiltration attack."

Okay. So in a sense they're cheating. Or at least they're modifying the rules in a Kobayashi Maru-like way. They're allowing for malware to rewrite the NIC's firmware to take control over the LEDs. In that case it would indeed be possible to hugely increase the rate at which data could be exfiltrated from an air-gapped network which has no other means of communicating, but those NIC cards can be seen.

What I appreciate I think most about these guys is that in every case they really do wrestle to the ground whatever wacky topic and method they are researching. They really do the work. For example, in this case their eight-page paper described the three methods which can be employed to control the LEDs of NIC interfaces. They said, okay, first, driver/firmware control. They said: "In this method, the LED-controlling code runs as a kernel driver or within the NIC firmware. Changing the LED state/color requires direct access to low-level registers or special non-volatile memory addresses. This method enables the highest degree of control over the LEDs, but is very hardware-specific and mostly undocumented.

"For example, documentation discusses how to control the Ethernet LEDs in an Intel NUC PC. It can be done from a kernel driver or by writing to specific addresses in flash memory at word 0x18, which holds the LED's configuration. For embedded controllers, the control of the NIC is typically performed via internal bus or USB interfaces. For example, sample code for LAN915X Ethernet controllers programs the corresponding LED register via USB commands." So that's the first way, and the best if you can get it.

Way number two: Link status control. They say: "In this method, only the status LED can be controlled. The malicious code can intentionally change the link speed, which in turn causes the network adapter to change the status LED. For example, setting the link speed to 10Mb, 100Mb, and 1Gb will set the status LED to off, green, and amber, respectively. Selecting the link speed can be done by interacting with the NIC driver. For example, the ethtool command-line tool in Linux enables to change the link speed of the Ethernet controller. The same is possible in the Windows OS via the netsh command.

"Note that setting the link speed requires root/admin privileges in both the Linux and Windows. Technically, the link speed is determined through the auto-negotiation procedure. In this procedure, which occurs in the physical layer, the connected devices share their capabilities regarding supported parameters such as transmission rate, half/full duplex, et cetera. The link speed of a network NIC can be determined from the computer's OS."

And third, user LED control. They say: "In this method, the user directly turns the status LEDs on and off by enabling and disabling the Ethernet interface using API or tools such as the ethtool or eth command. The user directly turns the status LEDs on and off by enabling and disabling the Ethernet interface. Another technique to blink the status LED is using the 'test' or 'identify' functionality, enabling the operator to identify the adapter by visual indication. These operations can be triggered programmatically or via low-level tools such as ethtool."

Okay. So as they always do, some of them, some group of them, whomever, really looked at this, and wrung out every detail, and actually implemented these strategies. They consider the cameras needed to receive the information, and the camera frame rates and interactions thereof, the maximum distances at which cameras can be focused upon LEDs on NIC adapters, and basically what can be done to make the entire thing work. Then they finally get down to the effective bitrates which are achievable through each of these three methods.

They have the driver/firmware control, that first and best one. They have in their table OOK, which is their short for on/off keying. And they say blink frequency and colors allows them to achieve 100 bits per second. So they talk about being able to use that to exchange text files, usernames/passwords, encryption keys, and PIN codes. The second approach is link status control. They can get 1 bit per second there. So you're not going to exchange anything really long, or at least not quickly. And then the final, the user LED control. They're claiming 2 bits per sec. So you could do keylogging, usernames and passwords, credentials, encryption keys, and so forth.

So that's the first, this blinking of NIC LEDs. You know, again, I discounted it because, again, it's not the actual data that's moving through the lines. But yeah, if you had control over the link, if you got software into the system in the first place, in a system that was unable to communicate with the outside world, but you were able to briefly infiltrate it, then you could clearly, given enough time, exfiltrate data. And really, if you could cheat the firmware and get 100 bits per second, then you could clearly do some damage.

But it's worth remembering that for many important secrets, you do not need a lot of bandwidth. Some of the very best kept secrets are also very short. A server's elliptic curve private key might only be 256 bits long. And even a larger RSA key is still only 2 or 4Kb. So even at a measly 1 bit per second, sluggishly bringing a LAN link up and down, a 2048Kb key can be transmitted in only a little over half an hour, about 34 minutes. So it's possible, if you wanted to do it.

Okay. So that's the first wacky idea. Wacky Idea #2, and arguably somewhat less wacky, their recently published 11-page paper is titled "GAIROSCOPE," and they spelled it weird, but understandably. They spelled it G-A-I-R-O scope, as in air-gapped, gyro, G-A-I-R-O scope. They said: "Injecting Data from Air-Gapped Computers to Nearby Gyroscopes." And you might think "Gyroscopes? What?" But they're in every one of our smartphones.

The paper's abstract explains. They said: "It's known that malware can leak data from isolated, air-gapped computers to nearby smartphones using ultrasonic waves. However" - and that was like from speaker to microphone; right? We talked about that little deal, and that was theirs, years ago. So "It's known that malware can leak data from isolated

air-gapped computers to nearby smartphones using ultrasonic waves. However, this covert channel requires access to the smartphone's microphone, which is highly protected in Android OS and iOS, and might be non-accessible, disabled, or blocked.

"In this paper we present 'GAIROSCOPE,' an ultrasonic covert channel that does not require a microphone on the receiving side. Our malware generates ultrasonic tones in the resonance frequencies of the MEMS gyroscope." MEMS is the abbreviation for Micro-Electro-Mechanical System, which is what these little itty-bitty, well, electromechanical systems are. So they said: "These inaudible frequencies produce tiny mechanical oscillations within the smartphone's gyroscope, which can be demodulated into binary information. Notably, the gyroscope in smartphones is considered to be a 'safe' sensor that can be used legitimately from mobile apps and javascript. We introduce the adversarial attack model and present related work. We provide the relevant technical background and show the design and implementation of GAIROSCOPE.

"We present the evaluation results and discuss a set of countermeasures to this threat. Our experiments show that attackers can exfiltrate sensitive information from air-gapped computers to smartphones located a few" - actually up to eight - "meters away" - more than 24 feet away - "via a Speakers-to-Gyroscope covert channel."

So this one is more serious and interesting. We were just talking about resonances last week with the Janet Jackson Rhythm Nation video and the Tacoma Narrows bridge. What's special about resonance is that a relatively small signal like a gust of wind up the Tacoma Narrows which would be entirely harmless in isolation, can sum into the power of successive properly-timed bits of energy to result in a significant signal. That's the effect these guys have taken advantage of here.

They explain what's going on in the MEMS gyroscopes. They said: "It is known that acoustic tones degrade MEMS sensors in a frequency range known as the 'resonance frequencies.' This ultrasonic input produces erroneously low-frequency angular velocity readings in the X, Y, or Z directions. The vulnerability of MEMS sensors to ultrasonic corruption is due to the mechanical structure of a MEMS gyroscope. The misalignment between the driving and sensing axes is one of the main causes of the fault output generated by the gyroscope.

"The phenomenon and its physical and mechanical roots are discussed in relevant literature. It was observed in the previous works that the typical resonance frequencies of MEMS are within a fragmented band in the ultrasonic frequencies mainly above 18 kHz. The frequency of the resulting vibrations within the sensor is determined by the structure of the MEMS gyroscope, its positioning, and the distance from the sound source."

As always, they do all the footwork, actually develop and implement a full attack from the software in the PC to run the speaker as an ultrasonic sound source, and determine how far away they're able to position the smartphone and what data rates they're able to get. They determine the natural resonance frequencies for a number of the MEMS gyroscopes and a number of different smartphones. And they demonstrate the ability to send 8 bits per second of binary data completely covertly and obviously silently because it's above our ability to hear from a standard PC speaker to a smartphone located up to 8 meters away. Actually I said 24. It's actually 26 feet. So that's pretty slick.

As always when I talk about the work that these guys are doing, I'm left thinking that it would be a blast to be in this professor's class, being asked to actually make these out-of-the-box attacks work. That would be an awful lot of fun. If you're not working for the U.S. government doing attack and defend and infiltrate or whatever that was.

Leo: We call it "cyber."

Steve: Cyber, yes. If you're not in the U.S. Cyber.

Leo: I just think of Donald Trump saying his son Baron was "excellent in the cyber." That's what I think of when I think of that. It would be a good class to be in, coming up with this stuff.

Steve: Lot of fun.

Leo: You could do it yourself. Maybe you should - let's think of a new way to do it. We've done glass windows. You said potato chip bags. That's for listening to audio. Exfiltrating from computers. Oh, there's got to be lots of ways. What about the sounds of the hard drive?

Steve: Oh, well, I mean, remember we even had the sounds of the power supply at one point.

Leo: Yeah.

Steve: And getting fans to spin at different speeds so like it speeds up and slows down.

Leo: Exactly.

Steve: And you can hear that at a, I mean, there's lots of ways. I mean, and remember the original, what was the military where they were able to, from a distance they were able to look at the electromagnetic noise coming off of a CRT screen.

Leo: Right. That's Tempest, yeah.

Steve: Tempest, that's the thing. Yup, Tempest.

Leo: They can do it through walls. Unbelievable. Unbelievable. Well, a fascinating story. Thank you for bringing that up. I appreciate it. This is why you listen to Security Now!; right? Mm-hmm. Get all the security news and some mind meat.

Steve: Mind candy.

Leo: Mind candy. Better than meat. Steve does Security Now! every Tuesday, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. Watch us do it live at live.twit.tv. There's audio and video streams there. Chat with us live at irc.twit.tv or in the Discord chat room. But you don't have to watch live. I mean, that's the whole point of, you know, we

record these. And we carefully craft a podcast out of them. We take this clay, and we shape it, and then we put the podcast up on the website.

Now, Steve's site is GRC.com. He has 16Kb versions of the show, that's a unique version, for the bandwidth-impaired. We do 64Kb audio, as well. He does it, and we do it. He also has transcripts, which really are handy for searching or for reading along as you listen, thanks to Elaine Farris, who does those for you every week, Steve. All of that at GRC.com. While you're there, you might want to take a look at SpinRite, the world's best mass storage maintenance and recovery utility.

Steve: Getting there.

Leo: Getting there is the word. SpinRite 6 is the current version. But if you buy that, you'll automatically get upgraded to 6.1, and you can participate in the rapidly winding down process of developing 6.1.

Steve: There actually will be a long beta period.

Leo: Oh, okay.

Steve: Because I'll have the DOS side code nailed well before I have everything packaged up as the Windows app that's able to produce the bootable USB sticks and everything else. So there will be something to owning 6.0 and being able to get the beta of 6.1.

Leo: Excellent. So it really is worth joining, then, or buying. Go to GRC.com. You can leave feedback for him there, GRC.com/feedback. You can also Twitter him. He is the tweeter guy, @SGgrc. We have copies of the show at our website, which is TWiT.tv/sn, 64Kb audio and video. That's our unique format, since we record video of this.

After the fact, it'll be up there usually a couple of hours, maybe three hours after the show. And you could also get it on YouTube. There's a dedicated YouTube channel to this show and all of our shows. And probably the easiest thing is subscribe in your favorite podcast player. That way you'll get it automatically, as soon as it's available. Discussions continue. Obviously chat is a little late, you know, if you're listening to a download. But we have a great forum. Steve has his forums at GRC.com. We have the TWiT Community Forums at TWiT.community. There's also a TWiT Mastodon. So if you want Twitter without the tweets, all the toots, none of the tweets, it's at TWiT.social is our Mastodon instance.

That concludes this thrilling, gripping edition of Security Now!. Thank you, Steve. Have a great week.

Steve: Thank you, my friend. I will see you in September.