The Mixed Blessing of a Crappy PRNG

Description: How long did it take for Windows' recent horrific WiFi flaw to be weaponized? What are the implications of the U.S. Commerce Department's total ban on Kaspersky? How is the Kremlin reacting? Why would an EU privacy watchdog file a complaint against Google for their Privacy Sandbox? When is an email tracking bug not a tracking bug? What can this podcast do to help a well-known security researcher present his work at DEFCON and Black Hat this summer? What's another near certainty for Microsoft's plan for Recall? What two mistakes have I been making on this podcast? And why might a really bad password generator wind up being a good thing?

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-980.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-980-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. Lots to talk about. The Commerce Department in the United States bans the Kaspersky Antivirus. Steve talks about why and whether it's a legitimate problem. We hear from a security researcher who is having trouble getting into the United States. Maybe you could help. And then we'll find out why every once in a while it's a good idea to have a bad password generator. All that and more coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 980, recorded Tuesday, June 25th, 2024: The Mixed Blessing of Lousy PRNG.

It's time for Security Now!, yeah, the show you wait all week for. Let's just make that your tagline. We wait all week for Tuesday. If it's Tuesday, it must be Steven Gibson day. Hello, Steve Gibson.

Steve Gibson: My friend, it's great to be back with you as we close out the first half of the year. And I have to say it was a little poignant, well, not poignant, it was very clear to me that we were closing in on Episode 1000, crossing the infamous 999, when I wrote 980.

Leo: Wow.

Steve: Because we're getting there.

Leo: Holy moly.

Steve: Yes. Okay. So I just have to say, and I assume you haven't seen it yet because you haven't fallen off your ball, that we have one of my favorite Pictures of the Week in a long time. So that's coming up. But I think a great episode, one with some interesting lessons: The Mixed Blessing of a Lousy PRNG. And I realized when I was using PRNG that some of our listeners might, what, a pring? Of course we know that's a pseudorandom number generator.

Leo: Yes. Do you think Pringles were named after pseudorandom - no, probably not.

Steve: Leo, that's it, exactly. This is the audience they were targeted at because we all sit around eating Pringles.

Leo: Yes.

Steve: But we're going to answer some questions, as we always do, before we get to our main topic, which is how long did it take for Windows' recent horrific WiFi flaw to become weaponized? Oh, and, oh boy, is there a new twist on this WiFi flaw, too. What are the implications of the U.S. Commerce Department's total ban on Kaspersky, which will be coming into effect in a few months?

Leo: Can you believe that? Wow.

Steve: Yeah.

Leo: Wow.

Steve: How is the Kremlin reacting to that? And who cares, but still. Why would an EU privacy watchdog file a complaint against Google over their Privacy Sandbox? Which is all about privacy, as the name suggests. When is an email tracking bug not a tracking bug? What can this podcast do to help a well-known security researcher present his work at DEFCON and Black Hat this summer? What's another near certainty for Microsoft's actual plan for Recall? This is something else that occurred to me that I think everyone's going to go, oh, of course, like the first time I had that thought a couple weeks ago. And what two mistakes - maybe not only two, but at least these two - have I been making on this podcast? And finally, why might a really bad password generator wind up being a good thing? A mixed blessing, as it were.

Leo: Yeah. I'm trying to think of why it could ever be a good thing.

Steve: And more importantly, what lessons do we learn about cryptography overall from that?

Leo: Yeah.

Steve: So I think, Leo, we may actually have a good podcast finally.

Leo: Well, after 980 attempts, I think it's good.

Steve: I think we got the hang of it.

Leo: We've kind of got it.

Steve: And this might be something where people come away thinking, you know, that was okay.

Leo: They're doing all right, these kids.

Steve: Yeah.

Leo: Well, let me tell you, before we go much further, about our first sponsor of the show. And then we can get into the...

Steve: I do have a penguin in my face, I should note.

Leo: Oh, I'm sorry. That's rude. How rude. I put a Linux penguin in your face. That's - sorry, you shouldn't go there. You go right here. Let me put him kind of off to the side. But just a little reminder that, if you're using Windows, you don't have to. Oh, by the way...

Steve: What we should all actually be using, that's right.

Leo: I found out what that was.

Steve: The thought that I have about Windows may cement that further.

Leo: Oh, boy. I found out what that was. Remember before the show Windows kept saying restart, restart, restart.

Steve: Yeah, yeah.

Leo: And then I was getting a UAC from 8bit Solutions. Turns out that's the Azure provider that Bitwarden uses. So that was Bitwarden asking me to update itself, basically.

Steve: Oh, interesting.

Leo: Which is, but, you know, I have to say, from a security standpoint, you don't want to see another name show up when you're reinstalling something. That means you have to go out and look it up and figure out why it wants to do that. So...

Steve: Well, and Leo, why is it only eight bits? I would think, what, is this from the '80s? 8bit Solutions? What?

Leo: I don't know. You wouldn't see it on any other platform.

Steve: Runs on a 6502 or something.

Leo: That's very odd, yeah. It's a good point. I don't know. Well, anyway, now I'm going to install it because I, you know, found out.

Steve: 128bit Solutions, and I'd be inclined to trust it.

Leo: More bits is better. Is it always better?

Steve: That's right, baby. If we learned anything in crypto, it's the more bits you've got, the better.

Leo: That's, what did you call that, padding of some kind; right? I can't remember, you had a good name for it. Password, Perfect Paper Password, Haystacks or something; right? Was it Haystacks?

Steve: Well, the Haystacks is an interesting idea, but that's different.

Leo: That's a different one, okay. All right. I am prepared to show you the Picture of the Week, soon as you say so.

Steve: And it is a fave.

Leo: I haven't looked at it yet.

Steve: It's just quick, visual, simple fun.

Leo: Okay. Do you want me to show it and then you describe it? Is that how you want to...

Steve: Okay, yes, that would be good.

Leo: All right. So I have to figure out how to show it first.

Steve: There it is.

Leo: You know, it's funny, my thing is moving around all the time, and I don't - sometimes my - now it's this laptop, so it's just going to take me a second. Why don't you set it up anyway?

Steve: Okay. So anyway, the picture. And I should note, Leo, that 4,330 subscribers to the Security Now! mailing list received this three hours ago.

Leo: Oh. So they're already up on it. They know all about it.

Steve: So email is live, yeah, and the email contained a thumbnail which a bunch of them clicked in order to see the full-size image. So anyway, I gave this the title "Correlation Is Not Causation" because...

Leo: That's a very important concept that people need to understand. Yeah, yeah.

Steve: It absolutely is. Yes. And we have a cute little, I'm not sure what he is, kind of a dog, but a small dog that's been leashed to a - isn't it wonderful?

Leo: He wants to get away badly.

Steve: He's looking at his master saying, hey, what about me? Why am I stuck here? So he's tied up to this, what would you call that?

Leo: A bollard.

Steve: A bollard, yes. However, something in the past has whacked this bollard off to the right so that it's a leaning bollard. And if you didn't know better, you'd think that this was Mighty Dog, and that in trying to join his owner he had tugged at this thing and yanked it almost out of the pavement. Anyway, I would commend our listeners to go find today's Picture of the Week because it is - it's a goodie. A lot of fun. And thank you to whomever it was who sent it to me.

Okay. So last week we opened with the news that the previous week's monthly Windows patch fest had quietly closed a remarkably worrisome flaw that had apparently been sitting undiscovered in every Windows native WiFi network stack since the last time Microsoft poked at it, and there's been no definitive statement about this because it appears that even Microsoft is quite freaked out by this one.

A listener of ours, Stephen CW, sent a relevant question: He said: "Hi, Steve. Longtime listener. Our corporate IT group vets Windows patches, thus delaying them. In the meantime, does turning off the WiFi adaptor prevent the attack you described?"

Okay, now, given the havoc that past mistakes in Windows updates have caused for corporate IT, especially remember a couple years ago when Microsoft kept wiping out all printing capability enterprise-wide, about like once a month they would do that? I suspect many organizations may have adopted a wait-and-test to avoid subjecting their users to such mistakes. And it's typically the case that even though 50 to more than 100 flaws may be fixed in any given month, nothing is really happening that's highly time-sensitive. But that's not the case with this month's revelations.

What I saw and mentioned last week at GitHub did not make any sense to me since it appeared to be too high-level. Remember, I mentioned in passing that there was already an exploit on GitHub. Well, since then, someone else appears to have found a way to overflow the oversized 512-byte buffer which Windows' WiFi driver provides for SSIDs. But that's not this problem.

He wrote, thinking that this was the critical 30078 CVE. He initially wrote: "CVE-2024-30078 describes a vulnerability in the way Windows handles SSIDs (Service Set Identifiers) in WiFi networks. Windows has a buffer for SSIDs up to 512 bytes long, which exceeds the WiFi standard. By sending chunked frames to increase the SSID size beyond 512 bytes, a buffer overflow can be triggered." He says: "This exploit leverages this vulnerability to cause a buffer overflow by creating and sending WiFi beacon frames with oversized SSID fields."

Okay, so that's a problem. But then he realized that he had found a different flaw from what Microsoft patched. So in an update he subsequently added, he said: "Info: This REPO does not seem to be hitting the same bug as in the stated CVE. New information has come to my attention thanks to FarmPoet. The CVE-2024-30078 vulnerability is in the function 'Dot11Translate80211ToEthernetNdisPacket().'" And I should say that absolutely, based on that function name, that makes total sense. And he said: "Of the native WiFi Windows driver (nwifi.sys), where a very specific frame needs to be constructed to get to the vulnerable code path," which, as he said, his code, his current code, does not. So he said: "I'm working on it. I've identified the changes in the patched function and am now working on reversing to construct the relevant frame required to gain code flow into this segment."

Okay. So we have this guy publicly working on a public exploit for this very worrisome flaw. And we're about to see why turns out this a lot worse than it first seemed. Meanwhile, it may be that anyone who has a spare $5,000 may be able to purchase a working exploit without waiting for a freebie on GitHub.

The online publication DailyDarkWeb - believe it or not there is such a thing - writes: "A threat actor has announced the sale of an exploit for CVE-2024-30078, a Remote Code Execution vulnerability in the WiFi driver affecting all Windows Vista and later devices. In their announcement, the threat actor details that the exploit allows for remote code execution over WiFi" - get a load of this, though - "leveraging compromised access points or saved WiFi networks." I'll get more to that in a second. "The exploit reportedly works by infecting a victim through WiFi router-based malware or simply by having the victim's device be within range of a WiFi network they've previously connected to.

"The exploit code is offered for sale at $5,000 U.S., with the price being negotiable. The seller also offers to develop custom solutions tailored to the buyer's needs. Anastasia, the new owner of the forum, is listed as the escrow for this transaction. Interested parties are instructed to send a private message to the threat actor, with a warning that time wasters and scammers will be ignored."

Now, in the first place, while we have no way to confirm this, from what we've seen before, it's entirely believable that several weeks downstream from the release of a patch which will have altered the binary of some WiFi component of Windows, that by "diffing"

- in hacker parlance - the pre- and post-patched files, the change that Microsoft made to repair the original driver's defect can be readily found. This is what the guy on GitHub is already doing.

But the really interesting attack vector that had not occurred to me when we first talked about this last week, but obviously has occurred to the author of this $5,000 for sale exploit, is the idea of infecting vulnerable consumer routers or corporate wireless access points, which might well be half the world's circumference away. In other words, if a vulnerable WiFi router is available anywhere in the world, it could be infected with knowledge of this critical Windows flaw, so that any unpatched Windows WiFi laptop within range of that router could be compromised. And that would be a very remote attack. It's clear that the only reason Microsoft was able to get away with labeling this flaw as only being "important," with a CVSS of 8.8, instead of "critical," with a CVSS of 9.8, or maybe even 10, is that it required a nearby attacker. At least that was the theory. But in reality, all it requires is a nearby hostile radio. And thanks to the historical vulnerability of consumer and enterprise routers, that's not a high bar.

The observation here is that a maliciously infected router may not be able to attack the machines connected to it by wire because there are no known exploitable vulnerabilities in their wired Ethernet network stacks. But that same router may now be able to successfully attack those same or other machines within its wireless reach thanks to the known presence of a, by Microsoft's own assessment, readily exploitable, low-complexity, highly reliable, likely-to-succeed flaw that exists in any Windows machine since Vista which has not yet received the patch that first appeared only two weeks ago.

So to answer Stephen CW's question about whether turning off, you know, disabling the WiFi on a machine will protect it, the answer has to be yes. Everything we know - although I have to say I looked around, and as I said, Microsoft is oddly mute on this whole thing. Normally you would expect them to say, mitigation, you know, disable WiFi. But maybe they presume so many people are using WiFi that you can't really call it a mitigation if taking the machine off the network is what it takes to mitigate the problem. So they're not suggesting that. But yes, everything we know informs us that turning off Windows WiFi adapters will completely protect any unpatched machine from the exploitation of this vulnerability.

Leo: Yeah. But you could also remove the machine from the Internet entirely, air gap it, and that would be good, too. I mean...

Steve: Or, Leo, something just hit me. Turn it off.

Leo: Yeah. That'll fix it, too.

Steve: What a concept. That's right. Pull the plug, shut it down, you're safe. Anyway, I wanted to conclude this week's follow-up on this CVE by making sure everyone understands that the addition of this remote router extension to this vulnerability really does change the game for it. We know tens of thousands of routers have already been and are taken over, and are being used for a multitude of nefarious purposes - launching DDoS attacks, forwarding spam email, as proxies to probe the Internet for other known weaknesses, and on and on. So the bad guys are going to realize that, by updating the malware that's already within their compromised router fleets, they'll be able to start attacking and hijacking any Windows machines that have not yet been updated that have their wireless turned on.

And for whatever reason, history tells us that there will be many such machines. Updating seems to be a slow process. And, for example, Stephen CW acknowledged that his corporate IT people, they're waiting now because there's been too much history of updates destroying corporate IT functioning. So they're taking a cautious process. Anyway, it's going to be interesting to see whether bad guys - how long it takes bad guys to leverage the idea of pushing this flaw out to the routers and seeing if they can remotely grab wireless machines.

I'll share this piece of news, this next piece, and interject some of my thoughts along the way. And Leo, I know you reacted a little bit as I did. And I'm of two minds. So it creates for some interesting dialogue.

Last Thursday, Kim Zetter, writing for Zero-Day, posted the news: "The U.S. government" - which did it on the same day. "The U.S. government has expanded its ban on Kaspersky software in a new move aimed at getting consumers and critical infrastructure to stop using the Russian company's software products, citing, of course, national security concerns. The ban, using new powers granted to the U.S. Commerce Department, would prohibit the sale of Kaspersky software anywhere in the U.S., and would also prevent the company from distributing software security updates or malware signature updates, to customers in the U.S." In other words, they're being cut off.

"Signatures," they explain, or Kim explains, "are the part of the antivirus software that detect malicious threats. Antivirus vendors push new signatures to customer machines, often on a daily basis, to keep their customers protected from new malware and threats as the vendors discover them. Without the ability to update the signatures of customers in the U.S., the ability of Kaspersky software to detect threats on those systems will significantly degrade over time.

"The U.S. Commerce Department announced the ban on Thursday after what it said was an 'extremely thorough investigation,' but did not elaborate on the nature of the investigation or what it may have uncovered," if anything.

"U.S. Secretary of Commerce Gina Raimondo told reporters in a phone call: 'Given the Russian government's continued offensive cyber capabilities and capacities to influence Kaspersky's operations, we have to take the significant measure of a full prohibition if we're going to protect Americans and their personal data. Russia,' she said, 'has shown it has the capacity, and even more than that, the intent to exploit Russian companies like Kaspersky to collect and weaponize the personal information of Americans, and that's why we're compelled to take the action we're taking today.'"

Wow. Okay. So in other words, we don't like their zip code, so we're going to deny a company, against whom we have no actionable evidence of wrongdoing, all access to the American market because, being a Russian company, they could be forced to act against us. And as I said, I'd say that I'm evenly divided on this. Through the years we've covered countless instances where Kaspersky has been hugely beneficial to Western software and to Internet security globally. Thanks to their work for the past many years, the world is a safer place than it would otherwise be. So to say "We don't like where you live so we cannot trust you" is a bit brutal. But at the same time, it is also understandable because, being in Russia, it's possible that their actions may not always reflect their values. And it's not as if operating within a state where we democratically elect our representatives is all that much different; right? After all, in the U.S. we have "warrant canaries."

Remember that Wikipedia explains a warrant canary by writing: "A warrant canary is a method by which a communications service provider aims to implicitly inform its users that the provider has been served with a government subpoena, despite legal prohibitions on revealing the existence of the subpoena. The warrant canary typically

informs users that there has not been a court-issued subpoena as of a particular date. If the canary is not updated for the period specified by the host, or if the warning is removed, users might assume the host has been served with such a subpoena. The intention is for a provider to passively warn users of the existence of a subpoena, albeit violating the spirit of a court order not to do so, while not violating the letter of the order."

So again, you know, state entities do. And in other words, in the U.S. our courts are able to say: "We demand that you turn over information within a certain scope; and, by the way, you're legally forbidden from disclosing that we've asked and that you have complied." So it's not my intent to pass moral judgment here. I'm just saying that what we see is unfortunately, you know, all nation states will act to protect their interests, and that their client citizens have little choice other than to comply.

So Kim's piece continues: "Asked what evidence the government found to support concerns that the Russian government is using Kaspersky software to spy on customers, Raimondo and other government officials on the call declined to provide specifics. One senior Commerce official said on background: 'In terms of specific instances of the Russian government using Kaspersky software to spy, we generally know that the Russian government uses whatever resources are available to perpetrate various malicious cyber activities. We do not name any particular actions in this final determination, but we certainly believe that it's more than just a theoretical threat that we describe.'" And that's right because these days, as we know, "belief" is all that's needed.

Kim writes: "The ban will not go into effect until September 29th to give existing Kaspersky customers in the U.S. time to find a replacement for their antivirus software. The ban on new sales of Kaspersky software in the U.S., however, goes into effect on July 20th. Sellers and resellers who violate the ban could be subject to fines from the Commerce Department and potentially criminal action. In addition to the ban, the Commerce Department also put three Kaspersky entities on its trade-restrictions entities list, which would prohibit U.S.-based suppliers from selling to Kaspersky, though it's unclear if Kaspersky currently has U.S. suppliers.

"A Kaspersky spokesman, in a statement to Zero Day, accused the Commerce Department of making its decision 'based on the present geopolitical climate and theoretical concerns..."

Leo: Well, yeah.

Steve: "...rather than on a comprehensive evaluation" - right.

Leo: I mean, I feel bad for Eugene Kaspersky. Everybody loves him.

Steve: Yes.

Leo: But why not? I mean, we don't have to have it; right?

Steve: Right.

Leo: And Huawei phones. I mean...

Steve: Right. I mean, this is what we're beginning to see happen; right?

Leo: Right.

Steve: As we choose sides, and we pull the bridges, the drawbridges up of global commerce that used to interconnect everyone. So anyway, they apparently think they have some legal standing. This spokesperson said: "We will continue to defend ourselves against actions that seek to unfairly harm our reputation and commercial interests." So, okay.

Now, as a little bit of background, the Department of Homeland Security had previously issued a directive in 2017 banning federal government agencies and departments - not consumers, and like everybody now, which is what is about to happen, so this was in 2017 - just federal government agencies and departments from installing Kaspersky software on their systems. DHS had also not cited any specific justification for its ban at the time, but media reports citing anonymous government officials back then cited two incidents, and we talked about them on the podcast. According to one story, an NSA contractor developing offensive hacking tools for the spy agency had Kaspersky software installed on his home computer.

Leo: Yeah, we reported this story. Remember this?

Steve: Yes, yes.

Leo: This was those NSA tools.

Steve: Right. He was developing these NSA tools, and the Kaspersky software detected the source code as malicious and extracted it from the computer, as AV software often does.

Leo: It quarantined it, yeah.

Steve: Well, it actually sent it to Kaspersky.

Leo: But that's what all antiviruses do. They quarantine it and send it in.

Steve: Exactly.

Leo: To the home office, which in this case was in Moscow. That was the EternalBlue leak.

Steve: Yes. So a second story claimed that Israeli spies caught Russian government hackers using Kaspersky software to search customer systems for files containing U.S. secrets. So, okay, you could install Kaspersky as you can many other tools. Mark Russinovich's PsExec is a favorite tool for bad guys to use, but its intention is benign. So Kaspersky for their part denied that anyone used its software to explicitly search for secret information on customer machines and said that the tools detected on the NSA worker's machine were detected in the same way that all AV software is designed to detect malware on customer machines.

Leo: Because it was malware. It really was malware.

Steve: Right, right, exactly. They were developing NSA malware for the NSA. And, you know, it's funny, too, because it's a little reminiscent of the Plex breach, which of course is the way LastPass got themselves in trouble. You have some, you know, some third-party contractor using your tools at home on their home machine, where they've got AV software installed.

Leo: Right.

Steve: It's like, whoops.

Leo: Whoops.

Steve: Not quite secure. So anyway, following that 2017 DHS directive, Best Buy and other commercial software sellers that had contracts with Kaspersky to sell computers with Kaspersky AV software pre-installed on those systems subsequently announced they would no longer install the software on computers they sold. This didn't, however, put an end to existing customers using Kaspersky software, or prevent new customers from purchasing the software on their own.

Today's ban is designed to convince those customers to stop using the software, as well. And get this. Commerce Secretary Raimondo told reporters: "When Americans have software from companies owned or controlled by countries of concern - such as Russia and China - integrated into their systems, it makes all Americans vulnerable. Those countries can use their authority over those companies to abuse that software to access and potentially exploit sensitive U.S. technology and data."

And I'll just note that the United States is no different in that regard. It's just that we're here, and they're there. We've covered the news that China's government is similarly urging its businesses to stop using Windows. We clearly have a new cyber cold war heating up; and unfortunately, choosing sides and cutting ties is part of the process.

So anyway, it's unfortunate, Leo, that a product that many people use is not going to be available. At the same time, I guess it feels to me like Kaspersky's employees should have seen the writing on the wall. They've seen the tensions between the U.S. and Russia heating up. It's easy for us to say, well, you know, they could have left Russia. But they probably love Russia as much as we love the U.S. So, you know, for most of them it's just a job.

Leo: And, I mean, Eugene Kaspersky was trained at the KGB technical school and did have a job in the Ministry of Defense when he founded Kaspersky Antivirus. So there are deep connections to the Russian government and the GRU.

Steve: And I would note also that AV software in particular has a very intimate relationship with an operating system. It is in the kernel. It is absolutely true, and this is the kind of thing that keeps the military mind up all night, you know, here we have a Russian company that has an active connection to all of the customers' machines in the U.S. And, you know, it's not a text editor. It's running a driver in the kernel.

Leo: This ain't TikTok. This is something else

Steve: Exactly.

Leo: Yeah.

Steve: If it did want to get nasty, in an instant it could take over all of the machines where it's installed.

Leo: And that's why it's been banned on government machines for a long time.

Steve: Since 2017, yes.

Leo: Yeah. So, I mean, yeah. It makes sense. I mean, I feel bad for Eugene. And part of the reason people are upset is everybody loves Eugene Kaspersky. Dvorak used to recommend Kaspersky all the time. He loved it. But mostly because he used to hang with Eugene and drink vodka during Comdex. So, but honestly, there are plenty antiviruses out there. One could argue you don't even really need an antivirus.

Steve: I was going to say, you and I, Leo, and my wife, and everyone I have any influence over, no longer uses any.

Leo: Right.

Steve: We just use Windows Defender. And believe me, it apparently is doing the job as it sure is a pain for me.

Leo: Oh, yes.

Steve: Let's take a break.

Leo: Oh, what a thought. My goodness. Yeah, okay. Good, okay. Yeah. Let me just pull the copy up here. I don't know, I'm a little off today.

Steve: Well, I'll just note while you're doing that that the Kremlin has extended the duration on its ban on Russian government agencies and critical organizations using IT services from unfriendly countries.

Leo: Yeah. That's exactly what's going to happen.

Steve: And that ban will enter into effect, the extension, on January 1st. So that was when the previous one was going to expire. And we still don't like each other, so you can't use those nasty Windows computers.

Leo: I mean, in a perfect world, I mean, I often think that the best way to keep from going to war is to have an economic dependency on each other.

Steve: Yes, yes. That's why it's like, it makes no sense for us to be upset with China. Everything we own comes from China.

Leo: Right. Right. And as a result, China's less likely to screw with us.

Steve: I would think so.

Leo: But maybe not. Who knows.

Steve: It's like, why are they messing with Taiwan, where the chips from? They're out of their mind.

Leo: But, see, that's more of a - see, this is the problem. That's not a rational thing. That's more an emotional thing. Just like Russia invading Ukraine because it used to be part of China. And so...

Steve: We want it back.

Leo: We want it back. But that's emotional. It's not rational, clearly.

Steve: Okay. So saw a short blurb in the Risky Business newsletter.

Leo: That's a good name.

Steve: And all it said was - sorry?

Leo: When you say "Risky Business," I think of Tom Cruise in underpants.

Steve: Yeah, I know. Always.

Leo: Always.

Steve: Okay. So it just was a short blurb. It said: "Google Chrome complaint." And it read: "European privacy organization noyb" - and it's not capitalized, so noyb? N-O-Y-B. It's Austrian - "has filed a complaint with Austria's data protection agency against Google for its new Privacy Sandbox technology. Noyb says Google is tricking users to enable Privacy Sandbox by falsely advertising it as an ad privacy feature."

Of course my reaction to that was "What?!" So I dug a bit deeper. I went over to the noyb.eu website and found their article with the headline: "Google Chrome: Agree to 'privacy feature,'" it has in quotes, "but get tracking."

Okay. So their piece begins with "After years of growing criticism over invasive ad tracking, Google announced in September of 2023 that it would phase out third-party cookies from its Chrome browser."

Leo: Wait a minute. Noyb stands for None of Your Business.

Steve: Oh, that is really good.

Leo: In German, I guess. Maybe not. No, it's English.

Steve: Really?

Leo: It's the European Center for Additional Rights, noyb.

Steve: That is perfect.

Leo: None of your business.

Steve: Okay. So this guy's saying: "After years of growing criticism over invasive ad tracking, Google announced it would phase out September of 2023 that it would phase out third-party cookies from its Chrome browser." So this is already misleading because, while it's true that Google has been using the same ad tracking that the rest of the advertising and data aggregation industry uses, the growing criticism has been over the entire industry's use of ad tracking, not just Google's.

You know, as we've been carefully covering here, what Google is hoping to do with their Privacy Sandbox is to change the entire model of the way advertising and its user profiling operates by inventing an entirely new way for a user's browser to intelligently select from among available advertisements that are seen at websites. And we've already heard from one of our listeners, whose job it is to implement the server-side technology

of a major website, that the rest of the non-Google industry is massively pushing back against Google's attempt to end tracking. Google really is trying to end tracking, and the rest of the community says no. We like tracking. We don't want you to take it away.

Okay. So the beginning of this article, I'll just share the beginning, it reads: "After years of growing criticism over invasive ad tracking, Google announced in September 2023 that it would phase out third-party cookies from its Chrome browser." That part I already read. "Since then, users have been gradually tricked into enabling a supposed 'ad privacy feature' that actually tracks people." Okay, it doesn't.

"While the so-called 'Privacy Sandbox'" - again in quotes from him - "is advertised as an improvement over extremely invasive third-party tracking" - which it is - "the tracking is now simply done within the browser by Google itself." Which is not true. "To do this, the company theoretically needs the same informed consent from users. Instead, Google is tricking people by pretending to turn on an ad privacy feature. Noyb has therefore filed a complaint with the Austrian data protection authority."

Okay. Now, the article goes on at length, and it never gets any more accurate. So there's no point in dragging everyone through it. It's full of misconceptions and an utter lack of understanding of what Google is trying to do. Google's Privacy Sandbox system explicitly does not track users, which is precisely why the rest of the well-established tracking industry is freaking out over it and scurrying around trying to come up with alternative tracking solutions.

This noyb is a privacy watchdog agency, as I said, based in Austria. I looked around their site, and they appear to gauge their value to the world by the number of complaints they're able to file per day. They're complaining about everyone and everything. So they're kind of like a rabid version of the EFF. You know, like the EFF, they are never going to be happy with anything short of complete and total legally and technically enforced Internet anonymity. And in a perfect world that would be great. But as we know, that's unlikely to happen.

Giving the author of this the most sweeping benefit of the doubt possible, the only thing I can imagine is that he confuses, hopefully not willingly, he confuses "tracking" with "profiling." Those two words are different, and so is what they mean. Perhaps he sees no difference. Perhaps he doesn't consider Google's Privacy Sandbox to be the "ad privacy feature" that Google does. We're told that websites which are able to offer identification of the viewers of the ads they display, or at least some reasonable assurance of the relevance to them of the ads, can double the revenue from their advertising. The problem, therefore, is not Google, who's been working long and hard to find a way to do this without tracking. The problem now is becoming websites and their advertisers who are refusing to change their own thinking.

Leo: It's challenging. By the way, you said IETF. Pretty sure you didn't mean the IETF, you meant EFF.

Steve: Oh. Oh, my god.

Leo: Because I don't think the IETF cares.

Steve: Of course. Thank you, thank you, thank you, Leo. You just saved me from receiving a thousand emails.

Leo: I know. "Steve, the IETF, the Internet Engineering Task Force, is not at all who cares about this at all."

Steve: Thank you so much.

Leo: I just thought I'd mention it.

Steve: Yes. Not the - I have a couple of errata coming up, so we just reduced the errata by one. Thank you.

Leo: I'm trying when I can, yes. Sometimes I miss it. I don't know. So there you go.

Steve: The EFF, exactly, thank you. But speaking of tracking, after last week's podcast, as planned, I finished the implementation of GRC's subscription management front end and turned to the email sending side. I designed a layout and template for the weekly podcast announcements I planned to start sending. And Saturday afternoon, U.S. Pacific Time, I sent the first podcast summary email to this podcast's 4,239 subscribers. And then, this morning, about three hours before - actually, we started an hour later than usual, so about four hours before this podcast began, I sent a similar summary of today's podcast to 4,330 subscribers. The list had grown by about 100 over the past week. So email is starting to flow from GRC, and everybody who has subscribed should have it. If you don't find it, check your spam folder because it may have been routed over there.

Rob in Britain said: "Hi Steve. As Apple broke their IMAP message read flag a while back, I've been using the Blue Mail app to get my email. Blue Mail includes a 'tracking image detector.' And guess what, it flagged your email message as containing one. As a Brit, the irony of a security podcast tracking me does not escape me."

Okay, now, Rob was one of a couple of people who replied with a "What the..." when their email clients reported that a so-called "tracking bug" was present in their email from me. And since that's what their client calls it, it's natural for concern to be raised. So I wanted to correct the record about when an email bug is tracking someone and when it's not.

The TL;DR is, it's not tracking you if it's a bug you indirectly asked for, and if it's only linked back to whom you asked from. The confusion arises because our email clients have no way of knowing that this incoming email is not unwanted spam, and that makes all the difference in the world about the purpose and implications of the bug because, if it were an unwanted spam email, as opposed to email everyone has been clamoring for, you would definitely not want your opening of that email to send a ping back to the cretins who are despoiling the world with spam.

But in this case, no one is being tracked because the image link points only back to me, back to GRC, the source of the email that was sent to you, which only those who jumped through some hoops to ask for it in the first place would have received. Also, unlike pretty much everyone else, and against the advice of some well-informed others, I (GRC) am sending the email myself, not through any of the typical third-party cloud providers that most organizations have switched to now using. As a consequence, the email address our subscribers have entrusted to me will never be disclosed to any third party. And as I noted, that single pixel "bug" is only coming back to me to allow me to obtain some statistics about the percentage of email I send that's opened and viewed.

And I've learned some interesting things, thanks to that little bug. For example, half of our listeners, well, I guess I already knew this already, half of our listeners are using Gmail. But I did not know that fully one quarter of our listeners are using Mozilla's Thunderbird as their email client. I thought that was interesting. So basically three quarters of everybody who has listed for email and received and opened their email from me, the last two that I've sent, three quarters are either - half of the total is Gmail, and the other one quarter is Mozilla's Thunderbird.

I'll also note that, as regards this bug, the Security Now! emails contain a link to the full-size Picture of the Week, and the show notes, and GRC's Security Now! summary page, all in the email, back to GRC. So it's not as if anyone who receives these emails from me and clicks any of their links is being stealth. Also, I chose to embed a reduced size Picture of the Week as a visible, about 250 pixels wide, thumbnail image so that the email would be self-contained and complete. I could have linked back to GRC for the retrieval of the thumbnail when viewed. And that way I would have obtained the same feedback that the single pixel image provided. And presumably, since it's, like, 250x203 pixels, it will just look like a real image, and it's visible, and no email client would say, oh, whoa, you've got a tracking pixel in your email. Right.

Anyway, it's certainly the case that unsolicited commercial spam email contains tracking bugs to inform their senders when their obnoxious unwanted spam has been opened by its recipient. Anyone who thinks that describes the weekly podcast summaries they signed up for will be glad that every one of my emails contains a very clearly marked unsubscribe link. And, of course, it has immediate effect. There's none of this "Please allow two weeks for your unsubscribe to be processed" nonsense. I've seen that from other, you know, major mailers. And I just think, wow, aren't you using computers?

Anyway, my work after today's podcast will be to automate the sending of these weekly podcast summaries. At the moment, sending a new email to a list is not difficult, but it does involve a large number of steps and decisions which are redundant week from week. So I want to take a bit more time to build some infrastructure to make it simple and mistake proof.

And Leo, I wish I had you. Are you nearby? No.

Leo: I'm coming. He can see me? I know, he knows I'm coming.

Steve: Yes, I do.

Leo: I tweaked my knee.

Steve: Oh, god, I heard about going up into the attic or something.

Leo: Yeah. I'm not used to stairs. And I fell up. I didn't fall down. I tripped.

Steve: So sort of like hit the front of your knee on the...

Leo: Yeah, I hit my knee, yeah.

Steve: Ooh.

Leo: It'll get better.

Steve: And of course stairs to an attic are probably not padded or carpeted.

Leo: Not good stairs, oh, no, oh, no.

Steve: So there are two things. We needed to take a break because we went a long time before our second one.

Leo: Yes, that's what I thought.

Steve: And you need to hear this next thing because this is from Orange Tsai.

Leo: Orange Tsai, yes.

Steve: The security researcher.

Leo: And you sent me a note about this.

Steve: Yup.

Leo: And I know what you're going to say. So, okay. Now, back to the show we go, Mr. G.

Steve: So I got an email with the subject "Seeking Assistance for Black Hat USA Visa Issue." And when I saw that this was from Orange Tsai, whose name should be familiar to all of our long-time podcast listeners, I thought, really? That one? I mean, that Orange Tsai?

So the email reads: "Hello, Steve Gibson and Leo Laporte. My name is Orange Tsai, a security researcher from Taiwan. While I'm not a listener of the show, Jonathan Leitschuh, a friend of mine, says you've featured my work and spoke about my name many times on the show. I've won the Pwn2Own championship and Pwnie Awards several times, as well as having been the researcher behind impactful research such as Exchange Server RCEs (mentioned in Security Now! 809, 819, 833, 844, 916 for ProxyLogon and ProxyShell), Samba RCE (Security Now! 857), Facebook RCE (Security Now! 795 for MobileIron RCE), SSL VPN RCEs (Security Now! 814 for FortiGate & Pulse Secure), and the recent PHP RCE." In other words, yes. We know Orange Tsai quite well on the podcast.

He says: "I come to you with a plea for support from either you or your listeners. I've been accepted to speak at Black Hat USA this year. Unfortunately, due to the United States border control, I've been unable to enter the country the past few years. I was

wondering if you or your listeners had any connections that would be of assistance in this.

"Here's a brief intro to give you some of the context: I've previously traveled to the U.S. seven times through ESTA (the U.S. Visa Waiver Program) and have presented in person at DEFCON and Black Hat USA many times without any issues. However, after I reported several critical vulnerabilities to Microsoft in 2021, my ESTA was rejected, my guess is because one of my reported bugs has a collision with a China APT Group. I believe this may have resulted in me being flagged by the U.S. Since then, I've been unable to enter the United States to present at DEFCON and Black Hat USA in person.

"In 2022, I tried applying for a business/tourist visa at the embassy. However, the consular officer couldn't decide, and my application had to be sent to DHS for further administrative processing. After several months of review, I never got a response and missed the 2022 DEFCON/Black Hat USA dates.

"This year, I submitted my latest research and was accepted by Black Hat USA in May of 2024," so last month. "To catch up with the visa this time, I reapplied for the B1/B2 visa in January and had the interview on March 19th. However, three months have passed, and there's still no update. As a security researcher, I try to do the most impactful research, and I'm keen on having my research seen by the world, especially at the top hacker gatherings like Black Hat USA. I'm currently seeking all the help I can get to break through this situation.

"I hope this gives you a better understanding of the situation I'm facing. This has been a long and troubling issue for me. If you have any advice or guidance to offer, it would be greatly appreciated. Here is my contact information in case anyone needs it. Thank you. Orange Tsai."

Leo: Wow.

Steve: Okay. So this is great, if we can help. And by "we," I mean everyone listening. So the moment I saw his name, as I said, my eyes opened wide because of course we recognized him from all the times we've talked about his many significant contributions to the security of this industry and its software systems. I don't actively maintain the sorts of contacts that he needs for this, like with the State Department. But I'm always surprised and flattered when I learn about the roles of many of the people who are listening to this podcast and who consider it to be worth their time.

So I'm sharing Orange Tsai's plea in the sincere hope that we do have listeners here who may have the connections required to solve this problem for him. This year's DEFCON and Black Hat USA conferences are being held near the start of August, and today is our last podcast of June. So we only have a month to go. I wrote back to Orange Tsai to tell him that I would be honored to do anything I could to help by giving his situation a larger audience. I also asked how someone who was in a position of authority might contact him if they needed further clarification.

He replied: "Hi, Steve. Thank you for your response. I really appreciate your help. My only concern comes via a friend, in that the U.S. government can be very sensitive to" - and he has in quotes - "'media pressure.' And there have been cases where this has led to a permanent ban on entry. Although Security Now! is not 'traditional media,' I still hope that, when mentioning my case, it can be done in a neutral manner. When seeking help, please ask listeners to do so in their personal capacity, rather than representing me, the media, or any other sensitive identities."

So anyway, I, speaking for myself, would ask anyone to heed that. You know, if you're in a position to help, please understand and be gentle if you're able to determine what's going on and why. I asked him for a link to a web page of contact information, which he provided. But all he wrote there was "Hi. I'm Orange Tsai, a security researcher from Taiwan. I really want to go to the U.S. to present my latest research at Black Hat USA 2024 in person. If you have any suggestions, please feel free to email me at orange[at] chroot.org. Thank you."

So with that, I'm leaving this in the hands of our wonderful listeners. You know, please don't do anything if you are not the right person. I would hate to make matters worse. But if you are the right person, or have a sufficiently close relationship with someone who is, then it would be wonderful if we were able to help him. His years of past work have shown that he is exactly the sort of security researcher whose work should be encouraged.

Markzip sent me a note. He said: "Hi, Steve. Seems to me that an overlooked problem with Recall is" - and this was interesting - "is third-party leakage. Listeners to Security Now! may lock down their machines and opt out of Recall, whereas the people with whom we interact may not. If I write an email to a friend, their Recall instance now knows of our correspondence. We can think of other leakage easily. For instance, people frequently share passwords via email. More examples should be easy to imagine."

Okay. So first of all, I think Mark makes a great point. Many people who've been critical of Recall have likened it to spyware or malware that's now being factory installed. Through our first podcast about this, well, I should say, although our first podcast about this was titled by me "The 50 Gigabyte Privacy Bomb," I have never characterized Recall as spyware or malware because both of those things require malicious intent, and at no point have I believed, or do I believe, that Microsoft has ever had a shred of malicious intent for Recall.

I've seen other commentators suggesting that the entire purpose of Recall is to eventually send the collected data back to Redmond for some purpose. I think that's irresponsible nonsense, and it's a failure of imagination. For one thing, Microsoft knows that in today's world they could never do that without being found out immediately. We are all now watching them too closely. And besides, why would they? The details of some grandmother's management of her canasta group is nothing that Microsoft cares about.

But that's not to say that there would not be some value to having the AI residing in Grandma's computer be "aware" of her interest in canasta. If Windows continues to evolve, or maybe devolve, into an advertising platform - which would be unfortunate, but seems likely based on the way it's going - Microsoft, think about this, Microsoft would be crazy not to use their Recall AI's digested history and understanding of its machine's user to improve the relevance of such advertising.

And as we know, this could all be done locally on the machine, much as Google's Privacy Sandbox will be doing in the user's web browser. In this case, the Windows OS itself would be pulling the most relevant ads from the Internet for display either in Windows itself or in their Bing web browser. So we now have one declared and two undeclared but obvious uses for Recall. And none of these applications for Recall's data requires it to ever leave its local machine environment.

The concern Mark raised about third-party leakage I think is a good one. It probably hadn't occurred to most of us that not only would our own machines be recording our every move, but that all of our personal interactions with any others would also be captured by their instances of Recall.

Last week we quoted Matthew Green on the topic of Apple's Cloud Compute design. He wrote: "TL;DR: It's not easy." And he said: "Building trustworthy computers is literally the hardest problem in computer security." He said: "Honestly, it's almost the only problem in computer security. But while it remains a challenging problem, we've made a lot of advances. Apple is using almost all of them." So that was Matthew talking about Apple's Cloud Compute. But the point being building trustworthy computers is the hardest problem we have.

So in Apple's case, they have the comparative luxury of housing their Cloud Compute infrastructure in data center facilities surrounded by strong physical security. Even so, the architecture Apple has designed does not require its physical security to hold in the presence of an infiltrating adversary. But they have physical access security nevertheless. That's something Microsoft does not have with their widely distributed Windows workstations. Grandma always leaves her Copilot+ PC open, logged in, and unlocked, just like her back door.

So Microsoft's challenge is greater than Apple's, which Matthew Green has just made clear is already the hardest problem in computer security. And as we've seen with last week's revelation of a super-critical WiFi proximity remote code execution flaw that's apparently been present in Windows forever, at least since Vista, whatever solution Microsoft finally implements will need to be something we've not yet seen them successfully accomplish.

Let me say that again because I think it's really important, and it's exactly the right way to think about this. Whatever solution Microsoft finally implements to protect its Recall data will need to be something we've not yet seen them successfully accomplish. What everyone else other than Microsoft clearly sees is just how much having Recall running in a PC raises the stakes for Windows security. But so far we've seen zero indication that Microsoft truly understands that this is not something they can just wave their hands around and claim is now safe for them to do because they said so.

What's not clear is whether they'll be able to use the hardware that's already present in those Copilot+ PCs to implement the sort of super-secure enclave they're going to need. And this is to your point, Leo, you made a couple weeks ago about that's really being what's necessary. And that makes it even more doubtful that they'll be able to securely retrofit the inventory of existing Windows 11 hardware to provide the required level of security. It may take new hardware. Apple has only managed to do it for their iPhone handsets because their hardware architecture is so tightly closed. Windows has never been, since it's an OS designed to run on third-party OEM hardware. So, for example, the phrase "Secure Boot" is an oxymoron since secure boot bypasses are continually surfacing.

I realize that I'm spending a great deal of time on Recall. This is now the fourth podcast where I've given it some significant discussion. And of course for the first two podcasts it was our main topic. But given the security and privacy significance of Microsoft's proposal, it would be difficult to give it more time than it deserves.

And finally, I have two pieces of errata. The first came from someone who wanted to correct my recent statement about the duration of this podcast. He noted that since we started in 2005, we are still in our 19th year of the podcast, not, as I have been erroneously saying, in our 20th year. So in two months we will be having our 19th birthday, not our 20th birthday. He said: "The reason we listen is that we know you care about getting the details right." I'm glad that comes through.

Leo: That's fair, yes.

Steve: So I'm happy to correct the record. And the second mistake several of our astute listeners have spotted is that I've been erroneously saying that the big security changes in Windows XP - its built-in firewall being enabled by default and its users' access to raw sockets being restricted - came about with the release of XP's final Service Pack 3. That's wrong. It was the release of XP's Service Pack 2 where Microsoft finally decided that they needed to get more serious about XP's security and made those important changes. So a thank you to everyone who said, "Uh, Steve." I appreciate the feedback.

Leo: Always. Wow, that's a deep cut. I mean, have you talked about that in a while?

Steve: Yeah.

Leo: Oh, okay.

Steve: The last couple weeks, actually.

Leo: Oh, all right. It's a long time ago. You can be excused for not remembering the exact details.

Steve: And I think the reason I was getting hung up on it is that I have had occasion to install some Windows XPs way later. And of course, after installing it, I always installed Service Pack 3, which was the last service pack, in order to bring it current.

Leo: But I do remember when they - it was a big deal when they built a firewall into Service Pack 2. That was like - in fact, I think we pretty much said, don't use XP until Service Pack 2 comes out, basically. It was a much-needed service pack, as I remember.

Steve: Yup.

Leo: All right. Well, you're forgiven.

Steve: I'm always happy to correct my mistakes.

Leo: Yes. That's good. Let us talk about something very important, our sponsor for this hour, and then get to the pseudorandom number generator, or PRINGLE, as I call it.

Steve: From hell.

Leo: For the PRINGLE from hell. Although if I'd only used it, I would be in heaven. So that's why it's a double-edged sword.

Steve: Actually, that's true.

Leo: Yeah.

Steve: You would be a little richer, as opposed to Little Richard.

Leo: Did you see it peaked at $67,000 a bitcoin?

Steve: Thanks a lot, Leo.

Leo: Don't do the math. Don't do the math.

Steve: Says he who formatted his hard drive.

Leo: 50. It might be worth, what, 3 million, almost 4 million.

Steve: North of, yeah. That hurts.

Leo: It's just money. Steve, money doesn't buy happiness.

Steve: I'll just have to earn it the old-fashioned way.

Leo: Yeah, there you go. Let's talk about PRNGs.

Steve: Yes. Leo, you may think it was bad. It's worse than you could have imagined.

Leo: Great.

Steve: So, yeah. "The Mixed Blessing of a Lousy Pseudorandom Number Generator," or "When are you very glad that your old password generator used a very crappy pseudorandom number generator?"

So today I want to share the true story of a guy named Michael who, after generating 43.6 bitcoin, lost the password that was used to protect it. With Bitcoin currently trading at around $60,000 U.S. for each coin, that's around $2.6 million dollars worth of bitcoin waiting for him at the other side of the proper password. Unlike many similar stories, this one has a happy ending. But it's the reason for the happy ending that makes this such an interesting story for this podcast, and offers so many lessons for us.

Okay, now, by pure coincidence, the story was recently written up by the same guy, Kim Zetter, who wrote that piece about Kaspersky for Zero Day that we were discussing earlier. Kim's story for Wired is titled: "How Researchers Cracked an 11-Year-Old Password to a $3 Million Crypto Wallet." He wrote: "Two years ago, when 'Michael'" - and

he has that in air quotes. Michael wants to remain anonymous because now Michael has a lot of money, and he would rather just keep it to himself. "Two years ago, when Michael, an owner of cryptocurrency, contacted Joe Grand to help him recover access to about $2 million worth of bitcoin he had stored in encrypted format on his computer, Joe turned him down.

"Michael, who is based in Europe and asked to remain anonymous, stored the cryptocurrency in a password-protected digital wallet. He generated a password using the RoboForm password manager and stored that password in a file encrypted with a tool called TrueCrypt. At some point, that file got corrupted, and Michael lost access to the 20-character password he'd generated to secure his 43.6 bitcoin, worth a total of about 4,000 euros, or $5,300, back in 2013 when it was generated and stored."

Leo: Yeah, a lot more now, baby.

Steve: That's right. "Michael used the RoboForm password manager to generate the password, but did not store it in his manager. He worried that someone would hack his computer to obtain the password." Reasonable concern.

"Joe Grand is a famed hardware hacker who in 2022 helped another crypto wallet owner recover access to $2 million in cryptocurrency he thought he'd lost forever after forgetting the PIN to his Trezor wallet," which is a hardware device. "Since then, dozens of people have contacted Grand to help them recover their treasure. But Grand, known by the hacker handle 'Kingpin,' turns down most of them, for various reasons.

"Grand is an electrical engineer who began hacking computing hardware at age 10 and in 2008 co-hosted the Discovery Channel's 'Prototype This' show. He now consults with companies that build complex digital systems to help them understand how hardware hackers like him might subvert their systems. He cracked the Trezor wallet in 2022 using hardware techniques that forced the USB wallet to reveal its password.

"But Michael stored his cryptocurrency in a software-based wallet, which meant none of Grand's hardware skills were relevant this time. He considered brute-forcing Michael's password, writing a script to automatically guess millions of possible passwords to find the correct one, but determined this wasn't feasible." Right, you know, 20 characters, upper and lower, special cases, numbers and so forth. As we know, 20 characters, that's strong security.

Leo: Believe me, I know.

Steve: Yeah, huh. That's right, Leo. "He briefly considered that the RoboForm password manager Michael used to generate his password might have a flaw in the way it generated passwords, which would allow him to guess the password more easily. Grand, however, doubted such a flaw existed."

Leo: [Laughing]

Steve: "Michael contacted" - uh-huh. The plot thickens. "Michael contacted multiple people who specialize in cracking cryptography. They all told him there's no chance of retrieving his money." And I should mention they should have been right. Right? You know, Joe Grand should have been right. All these crypto specialists should have been

right. "Last June he approached Joe Grand again, hoping to convince him to help, and this time Grand agreed to give it a try, working with a friend named Bruno in Germany who also hacks digital wallets.

"Grand and Bruno spent months reverse engineering the version of the RoboForm program that they thought Michael had probably used back in 2013, and found that the pseudorandom number generator used to generate passwords in that version, and subsequent versions until 2015, did indeed have a significant flaw." And let me just say calling it a "significant flaw" is like, you know, I don't know what.

Leo: It's understatement.

Steve: Calling noon "daylight" or something. I mean, okay. "The RoboForm program unwisely tied the random passwords it generated" - and I should explain I've dug down into the technology. I'm going to go into the kind of detail that our listeners want after I'm through sharing what Kim wrote. So he wrote: "The RoboForm program unwisely tied the random passwords it generated to the date and time on the user's computer. It determined the computer's date and time, and then generated passwords that were predictable. If you knew the date and time and other parameters, you could compute any password that would have been generated on a certain date and time in the past.

"If Michael knew the day or general timeframe in 2013 when he generated it, as well as the parameters he used to generate the password, for example, the number of characters in the password, including lower- and upper-case letters, figures, and special characters" - and by figures I guess he means numbers and special characters - "this would narrow the possible password guesses to a manageable number. Then they could hijack the RoboForm function responsible for checking the date and time on a computer and get it to travel back in time, believing the current date was a day in the 2013 timeframe when Michael generated his password. RoboForm would then spit out the same passwords it generated on the days in 2013. There was one problem: Michael could not remember when he created the password.

"According to the log on his software wallet, Michael moved bitcoin into his wallet for the first time on April 14, 2013. But he couldn't remember if he generated the password the same day or some time before or after that. So looking at the parameters of other passwords he generated using RoboForm, Grand and Bruno configured RoboForm to generate 20-character passwords with upper- and lower-case letters, numbers, and eight special characters from March 1st through April 20th, 2013. It failed to generate the right password. So Grand and Bruno lengthened the time frame from April 20th out to June 1st, 2013, using the same parameters. Still no luck.

"Michael says that Grand and Bruno kept coming back to him, asking if he was sure about this or that parameter that he'd used. He stuck to his first answer. Michael said: 'They were really annoying me because who knows what I did 10 years ago.' Anyway, he found other passwords he generated with RoboForm in 2013, and two of them did not use any special characters, so Grand and Bruno adjusted. Last November, they reached out again to Michael to set up a meeting in person. Michael said: 'I thought, oh my god, they're going to ask me again for the settings.' Instead, they revealed that they had finally found the correct password, no special characters. And it was generated on May 15, 2013, at 4:10:40 p.m. GMT.

"Grand wrote in an email to Wired: 'We ultimately got lucky that our parameters and time range was correct. If either of those were wrong, we would have continued to take guesses and shots in the dark, and it would have taken significantly longer to pre-compute all the possible passwords.'"

Kim then provides a bit of background about RoboForm, writing: "RoboForm, made by U.S.-based Siber (spelled with an 'S') Systems, was one of the first password managers on the market, and currently has more than 6 million users worldwide, according to a company report. In 2015, Siber (S-I-B-E-R) seemed to fix the RoboForm password manager. In a cursory glance, Grand and Bruno couldn't find any sign that the pseudorandom number generator in the 2015 version used the computer's time, which makes them think they removed it to fix the flaw, though Grand says they would need to examine it more thoroughly to be certain.

"Siber Systems confirmed to Wired that it did fix the issue with version 7.9.14 of RoboForm, released on June 10th of 2015; but a spokesperson would not answer questions about how it did so. In a changelog on the company's website, it mentions only that Siber programmers made changes to 'increase randomness of generated passwords,' but it doesn't say how they did this. Siber spokesman Simon Davis says that 'RoboForm 7 was discontinued in 2017.'

"Grand says that, without knowing how Siber fixed the issue, attackers may still be able to regenerate passwords generated by versions of RoboForm released before the fix in 2015. He's also not sure if current versions contain the problem. He said: 'I'm still not sure I would trust it without knowing how they actually improved the password generation in more recent versions. I'm not sure if RoboForm knew how bad this particular weakness was.'"

Kim writes: "Customers may also still be using passwords that were generated with the early versions of the program before the fix. It doesn't appear that Siber ever notified customers when it released the fixed version 7.9.14 in 2015 that they really should regenerate new passwords for critical accounts or data. The company did not respond to a question about this.

"If Siber did not inform customers, this would mean that anyone like Michael who used RoboForm to generate passwords prior to 2015, and are still using those passwords, may have vulnerable passwords that hackers can regenerate. Grand said: 'We know that most people don't change passwords unless they're prompted to do so.' He added that: 'Out of 935 passwords in my password manager,' he said, '(not RoboForm), 220 of them are from 2015 and earlier, and most of them are for sites I still use.' Depending on what the company did to fix the issue in 2015, newer passwords may also be vulnerable." We don't know.

"Last November, Grand and Bruno, having earned their reward, deducted a percentage of bitcoin from Michael's account for the work they did, then gave him the password to access the rest. The bitcoin was worth $38,000 per coin at the time. Michael waited until it rose to $62,000 per coin and sold some of it. He now has 30 BTC, now worth $3 million, and is waiting for the value to rise to $100,000 per coin.

"Michael says he was lucky that he lost the password years ago because otherwise he would have sold off the bitcoin when it was worth $40,000 per coin and missed out on a greater fortune. He said: 'My losing the password was financially a good thing.'"

Leo: Yeah, that's how I feel. If I ever - now, you can never recover yours. But if I ever remember my password, why, it's just been a long-term savings account.

Steve: Okay. So, first of all...

Leo: But a bad PRNG, ooh. They're always bad, aren't they? That's what the "pseudo" means.

Steve: Oh, well, Leo, wait for this. Oh, my god. First of all, RoboForm is probably a well-known name to everyone, even those of us who never had occasion to use it.

Leo: They were one of the first.

Steve: I'm in that camp. You're in that camp.

Leo: No, I think I used it, back in the day, though. I mean, many years ago.

Steve: Okay. Because it was the only - it was once the...

Leo: It was the first one, yeah.

Steve: Yes, yes. Okay. But since this podcast has been going since 2005, we've covered the span of time that RoboForm was apparently using a horrific password generation scheme. One of this podcast's early and continuing focuses has been on the importance of the strength of pseudorandom number generators used in cryptographic operations. So I was quite curious to learn more about what exactly Grand and Bruno found when they peeled back the covers of RoboForm circa 2013. And I was reminded of a line from the sci-fi movie "Serenity" where our villain says to Mel: "It's worse than you know," to which Mel replies, "It usually is."

Believe it or not, whenever the user of RoboForm v7.9.0 - and probably my theory is even v1, but we'll get to that in a minute. But definitely 7.9.0, which was released on June 26th of 2013. Whenever the user pressed its Generate Password button, RoboForm, up until its repair two years later with 7.9.14, simply took the Windows system's Unix time, which is the number of seconds elapsed since January 1st, 1970, and directly and deterministically used that time of day to produce the user's password. RoboForm didn't even take the trouble to create a unique per-system salt so that differing installations would produce differing bad passwords. This meant that if two users anywhere were to press the Generate Password button within the same one-second interval, if they were using the same password parameters, identical passwords would be generated.

Grand and Bruno discovered something else when they opened up RoboForm. The designers of this password generator, that should really just be called a "time scrambler," realized that if a user happened to press the Generate Password button a second time within the same second, the same password would be generated. To cover up this flaw, they subtract a fixed amount of time from the system time for repeats. What an utter disaster.

One thing we don't know is for how long RoboForm's password generator was this horrific before it was changed. I originally wrote "before it was fixed," but we don't know that it's been fixed. We don't know that it, you know, how it was changed. And we don't know why it was changed. But I have a theory about that. My theory is that this must have been the original implementation of RoboForm's password generator. The reason I think that is that by 2013 no one would have ever designed such a horrifically lame password generation scheme.

This had to have been a very early password generator created back in the late '90s or early 2000s before there was much awareness of the proper way to do these things. And then, following the well-understood property of software inertia, 10 to 15 years went by without anyone at RoboForm bothering to think about it again because it was, after all, producing random-appearing passwords. But for some reason, whatever reason, eventually someone noticed and apparently fixed it. We don't know how, but at least changed it.

Grand and Bruno note that something did finally change in 2015 with 7.9.14. But since RoboForm is both closed source and closed mouthed, we have no idea what may have precipitated the change, nor what the new algorithm was changed to. So I'm put in mind of Bitwarden, the password-generating sponsor of this network, where we can know anything we want to know about its innards, first because, if we ask, we'll be told; secondly because it's probably openly documented; and thirdly because the source code of the solution is publicly available. None of which is true for RoboForm.

The final note that's worth repeating is the point that Grand highlights: Regardless of their apparent complexity, we now know that's an illusion. It's just the scrambled time of day and date, without even having any per-system salt, which means that all user scramblings are identical for all owners of RoboForm, probably from the beginning, its first release, through 2015. Therefore, any passwords that were ever generated by RoboForm, presumably until 7.9.14, can be reverse engineered, and the set of possible passwords can be further narrowed by the degree to which their approximate date of creation is known.

Even if the format of the password is not known, there are a limited number of choices available for upper and lower case, special characters, numbers and length. So if someone were determined to crack into something that was being protected by a password that they had reason to believe had been generated by RoboForm, and they had some idea of when, such as the date of the protected account's creation, it's not a stretch to imagine that it could be done.

Sure, I would put the chances of this actually happening being done as extremely remote at best. But anyone who was using RoboForm back then, who may have never had the occasion to update their passwords since, should at least be aware that those passwords were simply generated by scrambling the time of day, and with a resolution of only one second.

Leo: That's terrible. That's so terrible.

Steve: There are not a cryptographically strong number of seconds in a day. And while I don't want to throw shade on RoboForm's products of today, which might be excellent, given the history that has just been revealed, RoboForm is certainly not something I could ever use or recommend, especially when there are alternatives like Bitwarden and 1Password which are hiding nothing, and RoboForm is hiding everything.

And this brings me to the final and most important point and lesson I want to take away from this. Way back when I and this podcast first endorsed LastPass, I was able to do so with full confidence. And in fact the only reason I was able to do so, and did, was because the product's original designer, Joe Siegrist, completely disclosed its detailed operation to me. It was the 21st Century, and Joe understood that the value he was offering was not some secret crypto mumbo jumbo. That was 20th-century thinking. Joe understood that the value he was offering was a proper implementation of well-understood crypto that was then wrapped into an appealing user experience. The value is not in proprietary secrecy, it's in implementation, maintenance and service. As we know,

many years and ownership changes later, LastPass eventually let us down. I hope Joe is relaxing on a beach somewhere, because he earned it.

Leo: I think he is.

Steve: So the lesson we should take from what can only be considered a RoboForm debacle is that something like the design of a password generator is too important for us to trust without a full disclosure of the system's operation and its subsequent assessment by independent experts. Any password generator that anyone is using should fully disclose its algorithms. There's no point in that being secret in the 21st Century. It doesn't necessarily need to be open source, but it must be open design. No company should be allowed to get away with producing passwords for us while asking us just to assume those passwords were properly derived, just because their website looks so nice. What the marketing people say has exactly zero bearing on how the product operates. It's obvious that we cannot assume that just because a company is offering a fancy-looking crypto product that they have any idea how to correctly design and produce such a thing. There's no reason to believe that there are not more RoboForms out there.

Leo: What's the best way, I mean, software random number generators are pseudo because they repeat eventually; right?

Steve: Remember that the first thing I started doing, the first piece of technology I designed for SQRL, and I talked about it on the podcast, was I created what I called an "entropy harvester." It was harvesting entropy from a range of sources. It was pulling from Windows' own random number generator. I fed mouse clicks and received network packets, DNS transfer rates, all the noise that I could was constantly being poured into a hash that SQRL was churning. And the idea was to create something unpredictable. Unpredictability is the single thing you want. And so the idea was that, I mean, like almost immediately SQRL's pseudorandom number generator would just have so much noise poured into it, all of that affecting its state, that there would be no way for anybody downstream to have ever been able to predict the state that SQRL's pot of entropy was in at the time that it generated a secret key.

Leo: Right. Gallia's reminding us that Cloudflare uses a wall of lava lamps to generate their random numbers.

Steve: Yes. Yes.

Leo: But it's not the seed you're generating because, as I remember with software random number generators, if you reuse the same seed, you'll get the same sequence of numbers. It'll repeat eventually; right?

Steve: Yes. Those are old pseudorandom number generators.

Leo: That's not how we do it anymore.

Steve: Right.

Leo: Okay. And I do remember you saying the best way to do it would be use a capacitor. Was that right?

Steve: Actually, a diode.

Leo: A diode, that's right.

Steve: A reverse bias diode where you put it just at the diode junction's breakdown voltage. And what happens is you get completely unpredictable electron tunneling across the reverse bias junction to literally create hiss. If you listen to it, it is hiss.

Leo: Right.

Steve: And it's truly - it is quantum level noise. And that's as good as it gets.

Leo: Wow. That would be the best way, you think? As good as it gets, yeah.

Steve: That is what all of the true random number generators now do. It is a variation on that. They actually do some post-processing because the noise can be skewed, but it is utterly unknowable.

Leo: That's actually fascinating problem in computer science because, you know, you might say, well, is a coin flip random? Well, it is with a perfect coin, but no coin is perfect. A roulette wheel is random with a perfect roulette wheel. But there is no such thing. They all have biases.

Steve: I was asked in 1974 to design a little machine that some people would take to Las Vegas. And it was going to be operated with toe switches because it could not...

Leo: Oh, that's the Eudaemonic Pie. This was in Santa Cruz; right? There's a book about this.

Steve: Actually, it was close, it was close to Santa Cruz, yes.

Leo: Yeah, there's this famous book about this. Have you read "The Eudaemonic Pie"?

Steve: No.

Leo: Well, they got caught. But they made a lot of money.

Steve: Yeah. And what they were doing was they were recording, at least in the case of the guys who asked me to develop this thing, they were recording roulette wheel results because no roulette wheel is perfect.

Leo: Right.

Steve: And so, and believe it or not, they had this thing running already, and they were using a wire recorder to record tones that their toes were generating, and they wanted me to do a solid-state version for them.

Leo: Yeah. They wore computers in their shoes to basically solve roulette, and they won a lot of money. And because, now, people are used to people counting cards in blackjack, but everybody in Vegas assumes, oh, a roulette wheel can't be beat. Well, they can. If you haven't read this book, you've got to read it. I wonder if it's the same guys. Very interesting story, "The Eudaemonic Pie." And I'm pretty sure that they were in the Santa Cruz area.

Steve: Well, that would be the right physical area because I was in Mountain View, which is just over the hill.

Leo: Yeah. And it was a toe computer. They would - wow.

Steve: Yup.

Leo: Wow. How fascinating is that? So, yeah, maybe someday, maybe if you've got, you know, a slow week - I know there's never a slow week on this show - you could talk a little bit about random numbers and why they're pseudo and why, you know, how to - it's a challenge. It's a nontrivial way to generate those with computers.

Steve: And crucially important. It's funny because we think about crypto as solving all the problems. But I'm not sure I can think of an instance where you don't need something random. When you're choosing a private key for public key crypto, you need high-quality random numbers. And we've seen failures of that where, for example, studies of the private keys used on web servers have turned up a surprising number of collisions of private keys because they were all getting their key shortly after turning on a version of Linux that hadn't yet had time to develop enough entropy. It hadn't warmed up its pseudorandom number generator enough.

Leo: I think that you were - I can't believe you've not heard of the book. The book focuses on a group of University of California Santa Cruz physics graduate students who in the late '70s and early '80s designed and employed miniaturized computers hidden in specially modified platform-soled shoes to predict the outcome of casino roulette games. I think you were an unwitting - you didn't do it, though; right?

Steve: I didn't do it.

Leo: You didn't do it. But they found somebody to do it.

Steve: Wow.

Leo: What a story. That may also be one of the first wearable computers, ironically.