



## A Chat with GPT

**Description:** All telecom providers have been hacked and may still not be safe to use. So now the government is recommending that we use our own encrypted communications. The plan to obsolete all non-TPM 2.0 PCs remains well underway. Microsoft must be feeling the heat, so they're taking time to not apologize. Whoops. Microsoft's product activation system has been fully hacked. All Windows and Office products may now be easily activated without any licensing. Here come the AI patents. Apple patents AI recognizing people by what they're wearing after earlier seeing their faces and noting what they're wearing. Zoom wasn't encrypting their early video conferencing. They're still trying to get out from under the mess their lies created for them.

High quality (64 kbps) mp3 audio file URL: <http://media.GRC.com/sn/SN-1004.mp3>

Quarter size (16 kbps) mp3 audio file URL: <http://media.GRC.com/sn/sn-1004-lq.mp3>

---

AWS introduces physical data terminal locations where users can go to perform massive data transfers to and from the cloud. The FTC has set their sights on data brokers. Let's hope something comes of it. GRC's email finally gets BIML. (Can you see the Ruby-G logo?) Lots of terrific listener feedback about authenticator policy, a new and free point-to-point link service, Tor's "Snowflake" linking PCs and Smartphones, and even recharging spent SodaStream canisters. Then we look at a recent conversation I had with "ChatGPT 4o with canvas" and the new plan that resulted.

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. His reaction to Microsoft's announcement you'll have to have TPM 2.0 for Windows 11, yeah, you might imagine Steve's a little upset. He'll talk about that in just a little bit. Apple patents AI recognizing people by the clothes they wear. The FTC is going after data brokers. And Steve's going to take a look at coding with ChatGPT. He has some very interesting thoughts. It's all coming up next on Security Now!.

**Leo Laporte:** This is Security Now! with Steve Gibson, Episode 1004, recorded December 10th, 2024: A Chat With GPT.

It's time for Security Now!, the show where we cover your security online, your privacy, your safety, what's going on in the world of cybersecurity, with a man, a plan, Panama. No, no, I'm sorry. That's wrong. Mr. Steve Gibson, that's who I'm talking about, of GRC fame. Hi, Steve.

**Steve Gibson:** Hello, Leo. Great to be with you for, what is this, Episode - like I don't know - Episode 1004.

---

**Leo:** It says it right at the top of the show notes. Big letters, 1004.

**Steve:** Hey, and what happens on the holidays? Because we have some - things collide with Tuesdays. I don't know if...

**Leo:** Oh, we are doing a Best Of for you; right, Anthony? I think we are. So there'll be a Best Of Security Now!. And in fact people can contribute their thoughts.

**Steve:** So there'll be a Best Of, and then we're off for a week; right? Or does the Best Of fill in for, like, the between Christmas and New Years?

**Leo:** Yeah, because Christmas is Wednesday. So Christmas Eve is probably when the Best Of is.

**Steve:** Yeah, and then New Year's Eve...

**Leo:** Anthony, as we planning to do a New Years Eve show, or not? No one knows. I will have to ask the boss.

ANTHONY: Can you hear me?

**Leo:** Yeah.

ANTHONY: We're dark.

**Leo:** We're dark.

ANTHONY: We're dark.

**Leo:** So no show on New Year's Eve.

**Steve:** Woohoo! Not - sorry, I didn't mean that.

**Leo:** You used to say, no, no, no, I've got to do a show. You don't care as much. Okay, that's...

**Steve:** Yeah, there's that wife again.

**Leo:** You have a life now.

**Steve:** I do.

**Leo:** You're going to be dancing.

**Steve:** I will be kept busy with endless social - what are we doing now, honey? Oh, well, we're going over...

**Leo:** Before you had a wife, you danced with James Tiberius Kirk on our set.

**Steve:** I was much more nimble back then, my friend.

**Leo:** That was - I'll never forget that. We did our 24-hour New Year's Marathon. Steve came up for it one year and danced with a cardboard cutout of the Captain.

**Steve:** Yeah, I don't remember - I remember I was either completely sober, or I was way far from it. I think actually I was completely sober.

**Leo:** You probably were.

**Steve:** And people would have thought that I was inebriated to be...

**Leo:** No, you were in the spirit.

**Steve:** ...making a fool of myself. But no.

**Leo:** No, it was fun.

**Steve:** Of course, on the other hand, you were getting your butt tattooed, so it was quite the event that year.

**Leo:** Mm-hmm. That all, by the way, video of that exists and is still available.

**Steve:** Don't. Don't, don't go - no. Don't, don't, don't.

**Leo:** I think it's on our YouTube channel. All right. What are we talking about more seriously, on a more serious note? What are we talking about today?

**Steve:** Okay. So today's podcast is titled "A Chat With GPT." And I listened to the end of MacBreak Weekly, where Andy and...

**Leo:** Alex was singing the praises of coding with ChatGPT.

**Steve:** Alex has had the experience. And it's interesting because he characterized it much as I have, that is, he was going further because he was wanting it to basically create an entire application framework with most of the things filled in except for a few things that, you know, it just couldn't get right. Anyway, I had an experience over the weekend that, you know, again, it was like, okay, what, what hap- what? And so I'm going to share that. And it is, I have to take us a little bit into the weeds of the questions that I was asking in order to set up the dialogue that we had, and then I have an announcement to make. But so this, again, today's episode titled "A Chat With GPT," but we have a lot of stuff to talk about.

All telecom providers have been hacked and may still not be safe to use, also which I heard you mention on MacBreak. So now the government is recommending that we use our own encrypted communications.

**Leo:** Uh-huh.

**Steve:** Uh-huh. Okay. Also the plan to obsolete all non-TPM 2.0 PCs remains well underway. Microsoft must be feeling the heat so they're taking the time not to apologize. Also, whoops, Microsoft's product activation system has been completely hacked, like fully. The things that the hackers weren't previously able to activate they can now activate. So all Windows and Office products may now be easily activated without any licensing. Also we're going to talk about the coming AI patents. Apple patented AI recognizing people by what they're wearing after seeing video of their faces and noting what they were wearing. So, well, that's an invention? Okay.

**Leo:** Yeah, hmm.

**Steve:** Zoom wasn't encrypting their early video conferencing, and they're still trying to get out from under the mess that those lies that they told back then created for them. AWS introduces physical data terminal locations at a metropolitan area near you where users will be able to go to perform massive data transfers to and from the cloud.

**Leo:** Bring your own hard drive. Just bring it on down.

**Steve:** Exactly. Also, the FTC fortunately has set their sights on data brokers, so we can hope that something comes of it. GRC's email finally gets BIMBI. And I had a lot of feedback from our listeners who received email from me showing our Ruby-G logo for the first time.

**Leo:** Yay. Yay.

**Steve:** So I'll update us on that. We also have a bunch of terrific listener feedback, one I'm going to go into some depth about authenticator policy and use. Also a new and free point-to-point link service, Tor's "Snowflake" proxy. Also a bunch of feedback from our listeners suggesting solutions for linking PCs and smartphones, which I have been complaining about my lack of ability to do. Also one listener said, "Steve, how do I refill

my SodaStream canisters again?" So I'll touch on that briefly. And then we're going to talk about the, like, shocking, well, to me because I was born in the mid-'50s, conversation I had with ChatGPT over some subtlety of assembly language syntax and how that went. So, and of course we've got a great Picture of the Week for our listeners. So I think a podcast that'll keep everybody entertained.

**Leo:** I'm very curious about ChatGPT's assembly language capabilities. That will be interesting. You know, it's actually a lot of controversy this year because people are using LLMs to solve the Advent of Code problems in seconds.

**Steve:** Oh, right.

**Leo:** It's immediately obvious because, I mean, even somebody's who's a professional coder, you know, competitive coder, will take a few minutes, at least, because you've got to look at the problem, read it, solve it, write the code. These guys are doing it in four or five seconds. It's obviously - they're using an LLM.

**Steve:** Wow. Well, and it's sad, too; right? I mean, like, okay.

**Leo:** What's the point?

**Steve:** If you enjoy playing chess, then why use a computer to cheat for you?

**Leo:** But people do that, even at the highest level.

**Steve:** I know.

**Leo:** And I know, it's very bizarre. I don't get it.

**Steve:** You know? So it's like me. I don't want it to program.

**Leo:** No.

**Steve:** I love to code. The reason, Leo, when I had 32 employ - or, no, 23, sorry, I got the digits backwards, I had 23 employees. I was upset because they were getting to have all the fun, and I had meetings. And I didn't want to have meetings.

**Leo:** That's no fun.

**Steve:** I wanted to do the R&D and write the code and worry about, you know, design ads and all that. And so now it's just me.

**Leo:** You get to do it all yourself.

**Steve:** With Greg and Sue to do the stuff.

**Leo:** Yeah, you don't want to do the bookkeeping or support.

**Steve:** No.

**Leo:** That part they can have, yeah.

**Steve:** Yes. They are my ChatGPTs in that case. But yeah. And so...

**Leo:** You know, it's probably only a matter of time before you've got an AI to do some of that stuff, too, yeah. Anyway, we'll talk about it. I'll be very curious what you found. So that's going to be fun. But that's just around the corner. I am ready, Mr. G. I haven't looked.

**Steve:** So this picture was a no-brainer to caption. It was demanding the caption "What could possibly go wrong?"

**Leo:** All right. I'm scrolling up now to see it for the first time. Oh. Oh, boy. That is a terrifying sight.

**Steve:** So, okay. Now, okay.

**Leo:** Sheesh.

**Steve:** I'm going to share the feedback that we've already received from one of our listeners.

**Leo:** Yeah.

**Steve:** Who asked ChatGPT what was the significance of this photo.

**Leo:** Oh, that's interesting.

**Steve:** So again, Leo, and this is - okay. So he wrote: "I, of course, also being blind, have taken to ChatGPT to get even better descriptions of images. I look forward to the upcoming show to see how it did, but I think you'd have to agree that the bar for image descriptions is now insanely high. It's certainly much better than anything we've had up until now, including Microsoft's and Apple's own vision models, useful as they've been in the past and being offline. Here's what I asked, and the response."

So our blind listener who received the email, there were 13,675 emails went out last afternoon to our listeners, he wrote: "Can you describe this in detail?" And then he gave it the URL, GRC.com/sn/1004.jpg, right, JPEG image. From that URL this technology we have today said: "The image shows a humorous but precarious scenario."

**Leo:** Oh.

**Steve:** "A red door with a silver handle is being unlocked using a key. The person holding the key is positioned above a metal grate that covers an open pit or vent below the door. The text above the image reads, 'This photo demands the caption, 'What could possibly go wrong?' The concern,' writes ChatGPT, 'is obvious. If the key slips or falls, it would likely drop through the gaps in the grate and into the pit below, making retrieval very difficult.'"

**Leo:** Wow.

**Steve:** "The situation highlights a classic Murphy's Law setup where the most inconvenient outcome seems inevitable."

**Leo:** Holy cow. That is really good.

**Steve:** Like what have - what? What? Astonishing.

**Leo:** I can only think that maybe, before we get too excited, that that image has shown up somewhere, in a newspaper, with similar dialogue below it, and that ChatGPT's making the connection. But that's pretty impressive. You know, my blog is hosted by a site called Microdot Blog. And it has an automated feature, because I post images there, of doing that, of asking an AI - because, you know, you want to, on a blog, if you have images, put an alt tag for unsighted readers like your correspondent there. And it does a very good job. I used to write my own alt tags, and I have to say this is a lot easier, to let the AI do it. But that is above and beyond. That's remarkable.

**Steve:** So I don't know that I need to further describe the picture because...

**Leo:** That's it.

**Steve:** ...ChatGPT just did.

**Leo:** It did.

**Steve:** One person commented that it might be a grate which is used in snowy country to allow people to scrape the snow, the packed snow off the bottom of their shoes. Although then I would think the grate would - the bars would be moving, would be oriented horizontally to make it more easy to scrape. I just think it was an inconvenient

location for a drain to be, you know, I mean, as we've seen, there are many instances where you wonder, okay, who's in charge here? This doesn't make any sense.

**Leo:** Yeah, yeah.

**Steve:** But anyway, it was a perfect setup for the topic we're going to get to at the end of the day.

**Leo:** Very impressive, yeah.

**Steve:** So, wow. And I shared our note with our also unsighted listener last week who thanked me for always going to lengths to describe the photos which she gets so much enjoyment from. So I just wanted to give her the tip that ChatGPT is standing by. And frankly, Leo, I'm going to - it'll be interesting to feed it maybe some more obscure images that seem less likely to have been, you know, populated on the Internet and just see if this was an anomaly. Or, again, it's just we have - there's something going on. And I don't want to step on my plan, but we'll get there by the end of the podcast. I have some news.

Okay. So Salt Typhoon is the name that's been given to this group. For the past several months there have been various news reports of Chinese state-sponsored attacks against this or that U.S. telecommunications company. I've seen them. I haven't mentioned them on the podcast because we've had so much else to talk about. And, I don't know, it sort of didn't seem to have reached critical mass. But that changed. Last week, Anne Neuberger, the U.S. Deputy National Security Adviser, said that at least eight U.S. telcos - and actually apparently a total of 80 overall - but eight U.S. telcos have been hacked, and that the U.S. is now getting set to take some concentrated definitive action. So I think we need to do a bit of catching up on the podcast.

The best reporting I found on this was headlined "Chinese hack of global telecom providers is ongoing, officials warn," with the subhead "Officials from the FBI and CISA say the major Chinese hack began late spring, and they're strongly, strongly urging Americans to use encrypted communications." Like, what? Okay.

Okay. So the reporting says: "Last Tuesday, federal officials said that the federal government began investigating a major Chinese breach of global telecommunications systems last spring, and they further warned that the intrusions remain ongoing, and that it's likely larger in scale than previously understood. The hack was first announced publicly in October and has been attributed by U.S. agencies to a Chinese government-linked hacking group known as Salt Typhoon. The effort targeted dozens of telecom companies in the U.S. and globally to gain access to U.S. political leaders and national security data. Neither the timeline of the hacking effort nor the scope of the intrusion were previously disclosed.

"Jeff Greene, executive assistant director of cybersecurity at CISA and a senior FBI official, said Tuesday that while agencies started cooperating on their investigations of Salt Typhoon's activities in early October, the effort was first detected in late spring and early summer. He also warned that the breach is ongoing, and that there was much law enforcement still did not know. Greene said: 'We cannot say with certainty that the adversary has been evicted.' Wow. 'We're on top of tracking them down, but we cannot with confidence say that we know everything, nor would our partners.' Greene strongly urged Americans to 'use your encrypted communications where you have it,' adding that

'we definitely need to do that, kind of look at what it means long-term, how we secure our networks.'

Wow. Yikes. That's definitive. And notice the irony of the government telling its citizens that they need to use their own encrypted communications apps wherever possible because the networks of the telecommunications providers are, well, turned out to be insecure, and there doesn't appear to be a lot that can be done about that. And we're not even sure we got rid of them, or what they're doing, or what's going on.

**Leo:** They're cockroaches. We can't get rid of them. They're in there permanently.

**Steve:** It's of course ironic, right, because our governments have been chafing over their citizens' use of the same encrypted applications which the government is unable to penetrate.

**Leo:** There's even more irony because the Salt Typhoon people are taking advantage of wiretaps that were inserted by CALEA 20 years ago because law enforcement said they needed them. The irony is endless.

**Steve:** Yup. So maybe as many as 80 - eight zero - telecommunications companies and Internet service providers, including AT&T, Verizon, and T-Mobile, are believed to have been infiltrated in the hack.

**Leo:** Eighty. There are 80 of them?

**Steve:** Eighty globally. Yeah. Basically all of them; right? Because we don't want to miss anybody with our CALEA warrant. T-Mobile was the most recent one in the news.

Anyway, earlier last Tuesday, CISA, the FBI, the NSA, and partner agencies in New Zealand, Australia, and Canada released a joint alert warning that Chinese hackers were targeting "major global telecommunications providers." Officials declined to comment on specifics, but acknowledged that "there were servers used in various countries to facilitate this activity by the Chinese."

Interestingly, the UK did not sign on to the alert, making it the only one of the Five Eyes intelligence-sharing group which was omitted. Greene attributed this to each country having "different considerations and timelines." Okay. A spokesperson for the UK's National Cyber Security Centre said Tuesday that the agency "supports our international partners issuing this advisory to help improve the collective resilience of telecommunications infrastructure," but at the same time didn't sign onto it. But oh, yes, we're supportive. We're just not going to put our name on it. And he also said the UK has a separate approach to mitigating cyber risks to its telecom providers. Okay.

Anyway, the officials from the FBI and CISA noted in their briefing that there were three groups of victims targeted in the hacks. The first group was an undisclosed number of victims, mostly in the "United States Capital Region," you know, meaning D.C.

**Leo:** Huh, D.C., huh? Hmm.

**Steve:** Yeah, huh, "according to the officials, who were impacted by stolen call records from telecom companies. The second group were a small number of political or government-linked individuals, all of whom have been notified by officials. So based on the records of this intrusion they at least were able to identify the targets of these attacks who had their private communications compromised, according to a senior FBI official who spoke anonymously as a condition of briefing the reporters.

"While the officials did not specify exactly how many officials were targeted, it was previously reported that the phones of President-elect Donald Trump and Vice President-elect JD Vance were among those compromised, in both cases prior to the U.S. national election. In many cases, the voice and textual content of call connections and conversations were obtained by Chinese attackers." In other words, not just metadata.

**Leo:** That's interesting. Well, it's a wiretap. So, yeah.

**Steve:** Yes, it was wiretap. "In addition, the Chinese hackers also accessed and copied U.S. court orders, which the FBI official said were attained through the, as you noted, Leo, Communications Assistance for Law Enforcement (CALEA) statute program. This program allows law enforcement and intelligence agencies to submit court orders around intelligence collection from telecom providers.

"When pressed on whether hackers were able to access court orders for intelligence collected under the Foreign Intelligence Surveillance Act (FISA) which allows U.S. intelligence agencies to collect data on foreign targets the FBI official declined to answer directly..."

**Leo:** Oh, god.

**Steve:** "...but acknowledged that the CALEA environment does include court orders for FISA investigations. The major hacking campaign has been an issue of increased concern for U.S. lawmakers in recent weeks, the Senate Intelligence Committee Chair Mark Warner describing it as 'the most serious breach in our history.'" Now, again, we installed the taps. So, gee, oops. I mean, this is like - isn't this the perfect analogy for why we don't want the government to have access to encrypted communications?

**Leo:** Yes. This is the whole proof.

**Steve:** They're just not - they're not good enough at it.

**Leo:** No one is. Any backdoor will eventually be discovered.

**Steve:** Yup. "Senator Mike Rounds, ranking member of the Senate Armed Services Committee's cyber subcommittee, said during a panel at last month's Halifax International Security Forum: 'Unless you are using a specialized app' - meaning, you know, our own encryption - "any one of us, and every one of us today, is subject to the review by the Chinese Communist government of any cell phone conversation you have with anyone in America.'" Okay. This is Senator Mike Rounds, you know, with the Senate Armed Services Committee's cyber subcommittee, saying unless you use something else, that is, just don't talk on the phone. Do something else. Unbelievable.

Anyway, I think this news highlights the clear need for independent third-party end-to-end encrypted video, voice, and text messaging systems. We're being told that the conversational content, not just connection metadata, of anything carried by our international and national telecommunications carriers can no longer be considered to be secure from eavesdropping by advanced persistent threat actors who want to know what's being said.

**Leo:** Well, they can have my phone calls. I'm not saying anything.

**Steve:** Well, right. But, you know, there are...

**Leo:** Still...

**Steve:** ...conversations which we don't want China to have. So if nothing else, this news, which has now been officially recognized, weakens any argument against allowing users of public telecommunication systems from providing and using their own truly secure end-to-end encryption for their conversations and content. The analogy is to the Internet; right? The Internet is a similar public network which is not, itself, secure. So to it we've added a layer of authenticated TLS encryption to enable point-to-point, end-to-end communications security (HTTPS), and no one has any problem with that. So what's the difference? And what's the big deal?

**Leo:** I should point out a reporter at Forbes looking at the actual request by the FBI that people start using encryption. The request said "Use responsibly managed encryption. Which is encryption that allows us to subpoena the cleartext."

**Steve:** Because we have responsibly managed telecom, and how's that working out?

**Leo:** So what they're saying is use encryption, but not too good. So, what, we should all use Signal or whatever it is that - Threema, whatever it is that you like. What would you use these days? Because you need to make phone calls, and it has to have audio as well; right?

**Steve:** Yeah. I guess I would use Signal if I had to have an end-to-end encrypted system that I trust. WhatsApp is using the Signal protocol so it's the same as Signal.

**Leo:** If you trust Meta. I mean, not sure I feel like trusting Meta, but okay.

**Steve:** Yeah.

**Leo:** Wow.

**Steve:** Yeah, I mean, certainly Signal doesn't have and can't have any other agenda because their entire, you know, business model is...

**Leo:** They don't even have a business model; right? I mean, what is their business? There is no business model. It's just...

**Steve:** Yeah.

**Leo:** What a world; huh?

**Steve:** Yeah.

**Leo:** It's just you've got to do what you suggested. Go out in the field. Take off all your clothes, go out in a field far away from any eavesdropping.

**Steve:** Get under a comforter with someone you, you know, don't mind being naked with.

**Leo:** And if you want to stay really private, bring them and...

**Steve:** Bring a space heater, a little portable, you know.

**Leo:** Yeah, make sure it's not made in China, though.

**Steve:** That's, oh, yeah. You don't want an Internet-connected space heater.

**Leo:** No.

**Steve:** It is the world we live in. At least...

**Leo:** There is no privacy.

**Steve:** No. No.

**Leo:** That's the sad fact.

**Steve:** Okay. So TPM 2.0, and we're not kidding.

**Leo:** Oh, we talked about this with Paul and Richard last week.

**Steve:** Wow.

**Leo:** Yeah.

**Steve:** A posting to the Windows IP Pro Blog last week was titled "TPM 2.0, a necessity for a secure and future-proof Windows 11." And of course I titled this bit of news "TPM 2.0, and we're not kidding." I'll give everyone a sense for this by sharing just the first few paragraphs of what is a quite lengthy posting by Steve Hosking, whose info on "X" identifies him as "Senior Program Manager for Windows Comercial." And I, you know, because I try to get my spelling correct, I noted that he has "commercial" spelled with one M. So, I don't - okay.

Anyway, he wrote: "With Windows 10 end of support approaching" - this is next October - "it's important to revisit a key minimum system requirement for Windows 11, Trusted Platform Module (TPM) 2.0. Let's discuss the role of TPM and its value for those of you who have made the transition to Windows 11. You'll also learn how to check your TPM status and how to prepare for Windows 11." Presumably for those who haven't yet transitioned. Ugh.

"TPM refers to a dedicated chip or firmware that offers hardware-level security services for your device. It securely houses encryption keys, certificates, passwords, and sensitive data, shielding them from unauthorized access. Additionally, TPM is tasked with cryptographic operations such as producing random numbers, encrypting and decrypting data, and confirming digital signatures. TPMs are available from many different manufacturers, including Microsoft on supported CPUs with Pluton." And I'll just note all of that's true of TPM 1.2 equally. Okay. But there's differences. We'll get to that in a second.

He continues: "You know that Windows 10 is approaching end of support. In Windows 11, TPM 2.0 advanced encryption techniques offer more versatile and critical key management for contemporary IT infrastructures, as compared to its predecessor, TPM 1.2. Integrating with features like Secure Boot and Windows Hello for Business, TPM 2.0 enhances security by ensuring that only verified software is executed and protecting confidential details. It's true that its implementation might require a change for your organization. Yet it represents an important step toward more effectively countering today's intricate security challenges."

And finally I'll finish with him saying: "TPM 2.0 helps keep your identities more secure and your data protection more robust. Can you ensure operating system integrity upon startup? Yes. Can you better protect sensitive information, data, and secrets? Yes. It provides a vastly more efficient and secure platform for Windows 11" - vastly, okay - "to use, through advanced encryption methods, improved industry standard cryptography, increased isolation, and greater interoperability with other security functions."

Okay. Enough of that. And then that's just like the tip of his iceberg. Okay. So is TPM 2.0 really better than 1.2? Yes, it is, without a doubt. It offers newer, updated cryptographic operations such as elliptic curve crypto and SHA 256-bit, SHA2-era hashing and message authentication functions instead of just SHA1. And it offers a privileged management hierarchy rather than just the single-level hierarchy, which isn't really a hierarchy, the single level offered by TPM 1.2.

But here's the problem. While 2.0 is without a doubt new and improved and should be adopted and used going forward, there's never actually been anything found wanting about TPM 1.2 that might force its abandonment. As we've observed from the beginning, this is an arbitrary requirement. TPM 1.2 had been working just fine for everyone, and still is, until Windows 11 came along. I would have no problem with Windows 11 taking advantage of the more secure features available from 2.0 if and when they were available in the underlying platform. But it should be up to Windows users whether or not

they feel they need to upgrade their PC hardware to obtain that additional security under Windows 11.

And Steven wrote: "It's true that its implementation might require a change for your organization." Right. A change. What he meant is that the move to Windows 11 may forcibly obsolete all of an organization's current stock of PCs which are otherwise, right now, still quite happily running Windows 10. But none of those machines will run Windows 11, and Microsoft's continuous IV drip of life-support to continuously repair the apparently endless supply of serious security bugs in Windows 10 will be coming to an end next October.

As we covered previously last Halloween, enterprises and individuals will have the option of paying for extended life support, for up to three more years in the case of enterprises, though it becomes increasingly expensive each year. Nevertheless, switching is always difficult. I get that. And I would not be surprised to learn that many of our listeners or their organizations were not seriously considering either paying to stay with Windows 10 on their current hardware, or perhaps switching to the arguably superior alternative offered by 0patch.

It rubs me the wrong way for Microsoft to be charging its customers to fix security flaws in its own products when it is already fixing them anyway, and has a well-running system in place that allows those fixes to continue being delivered. What Microsoft is planning to do next October is to deliberately disable the existing Windows Update for Windows 10 users who choose not to pay to have Microsoft continue to repair their own software flaws. What's wrong with this picture? As we noted last week, the United States government recently opened a broad antitrust investigation into Microsoft's abuse of its monopoly power. So Microsoft choosing to force the obsolescence of hundreds of millions of PCs - or hold their customers ransom over fixing those software flaws in their own products - could not come at a better time.

We've seen that it's possible for Microsoft to examine its own behavior and change when it's shown to be wrong. In the case of their cloud computing security, they were previously offering paid security enhancements through logging that should have been included at no charge as part of the base offering, rather than being disabled by default. Once it became clear that this conduct was unusual and wrong, they began including those additional services free of charge. October is still 10 months away, and there's time for another policy change regarding the future of Windows 10 and 11. Stay tuned. We'll see what happens.

**Leo:** Yeah. I think they will. We'll see.

**Steve:** It's just, again, they're making those updates available to people who pay.

**Leo:** Yeah. So that's - they're doing them.

**Steve:** And they're disabling, they're disabling that for Windows 11. Like [sputtering] I'm just - okay. Let's take a break so I can calm down, Leo.

**Leo:** Let's make sure that the break is not sponsored by Microsoft, and then - oh, yeah, we're good, okay.

**Steve:** I don't think we've ever had Microsoft as a sponsor.

**Leo:** I don't think we ever will.

**Steve:** I don't think that's going to happen.

**Leo:** That's right. Ain't gonna happen. They don't even sponsor Windows Weekly. But then again there's Paul Thurrott to deal with, so. And now back to Steve Gibson.

**Steve:** Okay. So while we're on the topic of Windows, or Microsoft, Martin Brinkmann, writing for gHacks, titled his piece "Hackers claim to have cracked Microsoft's software licensing protection almost entirely."

**Leo:** Oh, boy.

**Steve:** Uh-huh. He writes: "A team of hackers" - and it looks legit. "A team of hackers claim that they've cracked 'almost' - there's a quote - 'almost the entire Windows/Office software licensing protection.' The breakthrough allows them to activate 'almost any version of Windows and Office' permanently. Windows and Office installations require activation. This may happen behind the scene or when users enter product keys. Workarounds and hacks have been available for a long time. One popular choice requires running a single line of instructions from a PowerShell prompt to activate Windows 8 or later, or Office.

"The creators of the solution claim that they've found ways to extend this to even more Windows and Office products. The new method works on any Windows client or server version and includes Extended Security Updates, which Microsoft starts charging for next October unless they change their policy, and Microsoft Customer Specific Volume License Keys (CSVLK). The method used up until now could not activate everything permanently. But now, for the first time, the versions that had remained elusive have been supported: Windows 7, 8 and 8.1; any recent Windows Server; Add-ons; and Extended Security Updates are all added.

"The hack," he says, "for example, enables support for Windows 10 ESU, once it starts in October 2025. The hackers claim that the discovered method is simple. It does not require third-party file installations or system file modifications, according to a post on X." Okay, now, I've captured their posting to X which was posted by @massgrave, M-A-S-S-G-R-A-V-E. In this instance the reason they chose this moniker is MAS stands for Microsoft Activation Scripts.

And they posted: "Hi @everyone." They said: "We're thrilled to share some groundbreaking news from the @massgrave R&D team! Our team has successfully cracked almost the entire Windows/Office soft" - anyway, so they just repeat basically what Martin quoted them saying. For anyone who's interested, I have the X.com link to this posting in the show notes, and also the Powershell MAS scripts.

**Leo:** All right. Now, I'm about to do this, Steve. Should I be - is it scary? Is it nerve-wracking? Should I even do this?

**Steve:** Uh...

**Leo:** So I have a key. I installed a second virtual machine. And of course the product key was essentially the first one. I can go back and figure it out. But what if I just ran their little Powershell script here? What do you think? Oh, it said no. Okay. So maybe...

**Steve:** Actually it's complaining about not - something about SSL/TLS.

**Leo:** Not getting, yeah, not getting a TLS channel. I'm not sure why not. I'm online. All right. This was probably a foolish thing anyway. So I'm glad it stopped me.

**Steve:** Yeah, okay, well...

**Leo:** Right?

**Steve:** Okay. So...

**Leo:** I mean, it's downloading and running software from the Internet.

**Steve:** Actually, it is a local Powershell script that as far as I know does not need access to the Internet.

**Leo:** Oh, but this was - I was using that first option. So maybe it was fine.

**Steve:** Okay. Okay. So anyway, so I recognize this is controversial; right? But this is now not any secret. First of all, the scripts are hosted on GitHub, which Microsoft owns.

**Leo:** Oh, yeah, you're right.

**Steve:** And they're posting on X. When I looked, the first time I looked it had 913,000 views.

**Leo:** Wow.

**Steve:** Then I looked the next day, and it was 916,000, more than 916,000. So again, cat's out of the bag. I did download, because I was curious, I went to GitHub, Microsoft's property. Looked at their - and downloaded a zip containing their Powershell scripts. They look very comprehensive. They are very complex and detailed. You know, I didn't spend much time with it since I have no particular interest in any of this. I just wanted to report what has happened because it's news. And I'm sure that many frisky script kiddies out there, literally script kiddies, are already enjoying many hours of playing around with this to see what it does.

Martin finishes his reporting by writing: "An example screenshot of a fully, permanently activated version of Windows with Extended Security Updates has been shared as part of the post. The methods have worked for years" - this is Martin writing - "according to one of the follow-up posts. The hackers claim that their digital license method worked since 2018 and that the KMS method" - whatever those are - "for at least 17 years. The discovered hack will be made available in the coming months, according to the original post on X." So I'm a little confused by that because it looks like what's there is the whole deal. Maybe you're right, Leo. Maybe there is some piece of it that it's obtaining from the 'Net, although it looked like to me there was a lot of script there, a Powershell script that was doing all of the heavy lifting.

He writes: "The discovery is a serious blow for Microsoft, provided that the hack is indeed as foolproof and easy to apply as claimed. It's unclear how, or if, Microsoft will react to the hack. For now, it seems that the hackers have, at least temporarily, won the battle."

I'm not sure that I agree that it's a serious blow. You know, Windows is now free, essentially; right? I mean, it's loaded down with Microsoft crap that you get as part of it, and certainly they're being paid for the Start Menu to come preloaded with all of this junk. So there's that. Also, I posted the link over the weekend, Leo, and we know Paul Holder well. He related anecdotally his experience of reporting this to somebody, I mean, like years ago reporting this.

**Leo:** No, he knew about it.

**Steve:** And they just sort of shrugged. Like, you know, they know about it. They don't care. I think, you know, they figure, yeah, okay, well, we're selling bits that don't cost us anything. So if some of them get stolen, fine. You know, for my part I've been a paid-up Microsoft Developer Network (MSDN) developer for decades. I pay for the privilege of installing whatever Windows editions I need for software development and testing. But it is going to be interesting to see how this develops over time. You know, I never really...

**Leo:** I can't get it to run.

**Steve:** Okay.

**Leo:** Yeah, I mean, just I don't know what I'm doing wrong.

**Steve:** Well, Powershell scripts are finicky. You know, it may need some other module, or you may - did you right-click and run it as an admin?

**Leo:** Ah, maybe I need to do that.

**Steve:** There is that kind of thing, too.

**Leo:** Oh, I bet I didn't do that, yeah. Sure. Okay. I probably shouldn't do it on-air anyway because then there'd be video evidence of it.

**Steve:** Well, again, I don't know what the count current, what the current count is. I'm going to click on the link right now, and we're going to find out. The current number of views of that posting, last time I looked it was 916+ thousand. Okay. Now we're at 918.5 thousand views. So again...

**Leo:** Wow.

**Steve:** Not a secret anymore. And, you know, people are reporting that it works. So...

**Leo:** Very cool.

**Steve:** And again, you know, again, I never really thought about cracking the Activation System.

**Leo:** No.

**Steve:** But it's obviously been something of a preoccupation for some segment of the hacker community for quite a while. And, you know, again, it's like now you get Windows with any hardware that you buy. And if you set up your own hardware, I guess, what, I guess you have to pay a few hundred dollars for it. Or, you know, ask somebody else for their key or, you know, who knows. Anyway...

**Leo:** You can buy - you can buy keys online for pretty cheap, too. So, yeah, I think it's...

**Steve:** Anyway, just of interest for - and I thought I would report it because I'm sure that we have some parties among our listeners who will think, hey, this is cool. I'm going to do what Leo did, set up a VM and play with it, see what it does.

**Leo:** Yeah. I mean, the thing is I have a paid license. I just have to move it over, and this is easier than doing that.

**Steve:** Yeah.

**Leo:** Yeah.

**Steve:** Okay. So Apple was just granted a patent, like a week or two ago, with the title "Identity Recognition Utilizing Face-Associated Body Characteristics." And that serves to give some sense for where and how future AI will become packaged into consumer devices because this is an AI-based patent. The gist of the patent is that from the standpoint, if you think about it, of a fixed security camera, someone's face provides the most useful recognition detail. But the camera might not always be able to see the person's face.

So while the camera IS seeing the person's face and is able to identify them, it will also now, per Apple's patent, be taking note of other things - the clothes they're wearing at the moment, you know, that day, and their body dimensions, and their walking gait. Then that person may later be recognized, not by their face, which might not be visible at the moment, but by association with the other available characteristic details that had been previously noted at an earlier time when their identity could be positively determined.

So, okay. It's roughly the same sort of strategy that a human observer would employ. And the fact that the U.S. Patent and Trademark Office granted Apple a patent on this suggests that the AI revolution is going to further swamp the already buried USPTO as people apply for patents on a gazillion other seemingly obvious things that AI will soon be making commonplace. I have a link to this in the show notes for anyone who's interested. But it's like, I think the expression is "Katie bar the door" or something? It's like, wow.

**Leo:** You go all country on us. That's great.

**Steve:** Wow. I mean, it's like, you know, this has always been my problem with patents. There is a phrase in the law, in like in patent doctrine, that says that a so-called invention is not suitable for patent if anyone reasonably trained in the art would see this, if it would be obvious to anyone, reasonably obvious to anyone trained in the art. Meaning that, okay, is this like some flash of inspiration by a genius Apple developer? Or do they have bored patent attorneys in Cupertino who are saying, just give us something?

It's like, and people are saying, okay, how about this one? And oh, that's great, we'll write it up. We'll get a patent. You know, it is abuse of the system. While on the other hand, that's what patents have become; right? You build a portfolio as a defensive measure so that you're able to do things other people are doing. And when they say, hey, we've got a patent on that, you say, okay, yeah. But, you know, you're doing things that we're doing. So let's just agree not to sue each other, and we'll keep everybody else frightened. Wow. Okay.

**Leo:** Unbelievable.

**Steve:** I know. Mashable caught an interesting story last week. Their piece was titled: "Zoom lied about encryption in 2020. Now it wants to pay \$18 million to make that go away." And they tagged it with the subhead: "The Internet never forgets, though." Mashable wrote: "Back in 2020, Zoom was one of the hottest software companies in the world." And of course you and I were using them, Leo, because, I mean, COVID happened; right?

**Leo:** And it works. It works well. It's a good product.

**Steve:** Yeah, exactly. It works.

**Leo:** Yeah.

**Steve:** They wrote: "Its video conferencing software surged in popularity due to millions of people being confined in offices, home offices due to the COVID-19 pandemic. Unfortunately, the company cut some corners when it came to the privacy of its users. Despite Zoom's claims that its video meetings were end-to-end encrypted, it later came to light that this was not true. The result was a class-action lawsuit that Zoom settled for \$85 million. In 2021, Zoom also settled with the Federal Trade Commission over misleading its users about the privacy and security of its core product.

"But the matter did not go away entirely. There's also the separate matter of a U.S. Securities and Exchange Commission (SEC) probe into Zoom's privacy policies, which the SEC launched in 2020. Now Bloomberg reports that Zoom is offering to settle the matter with the SEC by paying an \$18 million fine. The offer is still pending approval by the SEC.

"These days, Zoom does offer end-to-end encryption for its video meetings, and its privacy and security practices have improved. But back in 2020, the company's track record was poor, with Zoom bombings" remember, "instances of people hijacking other people's Zoom calls and harassing them becoming something of a trend."

And the Mashable article finishes by noting: "By the way, if you've missed it, Zoom is no longer called 'Zoom Video Communications,' which was its official name until Monday. The company is now officially called Zoom Communications to reflect the fact that it now offers a suite of communications tools beyond its videoconferencing platform." And in fact one of them is a shared, a cloud Word competitor, you know, shared note-taking and document editing capability.

Anyway, we spent a lot of time talking and covering Zoom back during those explosive days, and we knew that its security was stumbling a lot during those early days. I recall that we talked about the "Zoom Bombings," as they were known, but I don't remember whether we actually knew that they were lying about, at the time, about their video conference calls not being truly end-to-end encrypted. Certainly it is challenging to do that. The easy way to do it is to encrypt to the hub, you know, a Zoom hub, so have each conference link encrypted to the hub, but then decrypt it there for redistribution and reencryption out to the other members of the video conference, which is presumably what they were doing. But that's not end-to-end. You know? That's, you know, they get to decrypt and then reencrypt. So that's probably what was going on. And, you know, if they're now doing it properly, that's a good thing.

So one of the problems posed by cloud services, especially in this era of "Big Data" - where "Big" can increasingly mean "really ridiculously humungously big" - is the question of how to seed the cloud by transferring massive amounts of data to and from a cloud provider who will, after that transfer, then become its host. To answer that need, Amazon has launched the first of their so-called "AWS Data Transfer Terminals." Here's what Amazon explained on December 1st under the headline "New physical AWS Data Transfer Terminals let you upload to the cloud faster."

They wrote: "Today we're announcing the general availability of AWS Data Transfer Terminal, a secure physical location" - like a Kinko's print shop - "where you can bring your storage devices and upload data faster to the AWS Cloud. The first Data Transfer Terminals are located in Los Angeles and New York, with plans to add more locations globally. You can reserve a time slot to visit your nearest location and upload data rapidly and securely to any AWS public endpoints, such as Amazon Simple Storage Service (Amazon S3), Amazon Elastic File System (Amazon EFS), or others, using a high throughput connection." And there they mean really high throughput.

They said: "Using AWS Data Transfer Terminal, you can significantly reduce the time of ingesting data with high throughput connectivity at a location near you. You can upload large datasets from fleets of vehicles" - they're just giving examples - "operating and

collecting data in metro areas for training machine learning models, digital audio and video files from content creators for media processing workloads, and mapping or imagery data from local government organizations for geographic analysis.

"After the data is uploaded to AWS, you can use the extensive suite of AWS services to generate value from your data and accelerate innovation. You can also bring your AWS Snowball devices to the location for upload and retain the data for continued use and not rely on traditional shipping methods. You can find the availability of a location in the AWS Management Console and reserve the date and time to visit. Then you can visit the location, make a connection between your storage device and S3 bucket, initiate the transfer of your data, and validate that your transfer is complete."

I got a kick out of this: "On your reserved date and time, visit the location and confirm access with the building reception. You will be escorted by building staff to the floor and your reserved room of the Data Transfer Terminal location. Don't be surprised if there are no AWS signs in the building or room. This is for security reasons, to keep your work location as secret as possible."

And, you know, this sort of thing makes sense after you hear it; right? Once these AWS terminals are available in many major metropolitan areas, it's easy to imagine them becoming quite popular. The other thing that occurred to me when they said, you know, don't be surprised if there are no AWS signs in the building or the room, is this could all be time-shared. Like all other cloud providers, you know, this could be a provision made by some entity which is creating a multiuse high-bandwidth access to the Internet facility.

And, you know, the other providers are also going to be announcing similar terminals. And, gee, what do you know? They're at the same physical location. And, you know, they may not just be AWS that is using that. I got the sense, I saw a photo of a long corridor with lots of doors and the sense that it might just sort of be a general purpose access to the cloud facility. So anyway, kind of a cool idea.

The grc.sc shortcut I created to quickly take people to that Pentester website, which you and I both used when we were talking about this, Leo, which it allows anyone to quickly check for their data among all of that which was leaked by the National Public Data breach, is the number one most clicked shortcut of all time. It's grc.sc/npd. And when I checked just now, that is, yesterday, it had been used 12,394 times since its creation on August 20th.

I should mention that this was in the show notes, which our listeners received, those who were subscribed to them, yesterday. One of them was reminded of this, clicked the link, and found that it was no longer taking them. It was taking them to Pentester.com, but unfortunately those guys at Pentester.com were unable to resist the temptation of monetizing the traffic that was generated. So the shortcut no longer takes you there. I'm not going to take people to the site when they've sort of done a bait-and-switch. So it just takes you to a page at GRC that says we're sorry, but these people were unable to resist the temptation of monetizing the traffic.

So anyway, my point here is, as I've observed since, unregulated data brokers - just by their very existence, just the aggregation of what is available on the Internet, that aggregation itself represents a clear and present danger to society at large. So I was glad to encounter the news that the U.S. Federal Trade Commission had taken regulatory action against two U.S.-based data brokers. The FTC has banned Mobilewalla, Gravy Analytics - what a name - and its subsidiary Venntel (V-E-N-N-T-E-L) from selling the geolocation data of their users, that is, of the data that has been aggregated.

The FTC cracked down on the three companies after they were caught collecting and selling the information they had aggregated without their customers' consent; right? It's very much the way we never gave the credit bureaus our explicit consent to, at least that we knew of, to be collecting our data. They just, you know, got it. The FTC said that the data contained information about military sites, churches, labor unions, and other sensitive locations; and the FTC specifically singled out Mobilewalla for selling geolocation data to identify women who visited pregnancy centers and individuals who attended George Floyd protests. So it's difficult to find any sympathy for such parasitic companies.

I should also note that, when I did a sort by the frequency of clicks on GRC's shortcuts, the second most popular GRC shortcut was `grc.sc/pin`, which our long-time listeners, actually you don't have to be a long-time listener because it wasn't that long ago, took us, yes, to that wonderful graphic heat map which clearly showed the extremely non-uniform distribution in the four-digit PINs chosen by those who use PINs. So just a reminder that we have a lot of fun on this podcast.

**Leo:** Two excellent shortcuts. Keep them both. Have them ready.

**Steve:** Okay.

**Leo:** I did that Pentester website, you may remember, and was dismayed by...

**Steve:** Yes, you and I both found our data. And significantly, Lisa's data was not.

**Leo:** Was not.

**Steve:** Because she had subscribed to a data scrubbing service.

**Leo:** Yup.

**Steve:** Okay. Last week I received notification from DigiCert that they had approved the use of GRC's "Ruby G" logo for display in GRC's BIMi-certified email.

**Leo:** I guess the Internet Archive came back up.

**Steve:** Yup. Our "BIMi up Scotty!" podcast was back on October 15th, and at the time the Internet Archive - which the entire industry uses for this purpose to verify the long-term use of corporate logos - was suffering a long-running and debilitating series of DDoS attacks. And I'm sure that if my need was urgent enough, I could have reminded DigiCert before now and pushed the matter. But, you know, they did get back to it on their own without me needing to do so.

When I checked last week after receiving that notice, the certificate's status was in "awaiting final" status. I'm mentioning this because I awoke yesterday to the news that GRC's Verified Mark Certificate - which was the goal of all this - had been approved and was issued. Although I could have hosted the pair of files. One's an SVG, a scalable vector graphic, and the other is a PEM, a PEM certificate from my GRC.com domain, I

decided that it might seem a little more official if those files came from DigiCert themselves, though I doubt that it matters either way. But since they were pleased to offer to host the files, I took them up on that offer.

So then yesterday, Monday morning, I added a BIMI TXT record to GRC's DNS which contained the twin URLs for GRC's logo image and its matching certificate. From that moment, GRC's received email was BIMI enabled. Since I didn't yet have this week's email ready, I sent last week's email to my Gmail account, since Gmail is one of the providers that supports the display of BIMI logos. And sure enough, there was GRC's "Ruby G" decorating the opened email. I imagine that everyone who receives this week's email, and all subsequent email from GRC through BIMI-supporting providers, will also receive the same thing, whether they notice it or not.

And actually that's confirmed now. I got a whole bunch of email from people. And I don't know if they looked at the show notes and saw me talking about it, or looked at probably just the synopsis where I do mention that, but a whole bunch of people wrote back and said, "I see it, I see it, I see it." So indeed. Although there were some people who didn't see it. And that's just because their email provider isn't yet showing it.

**Leo:** Yeah, most aren't, I think. Do you have to turn that column on in Gmail? Or is there some...

**Steve:** No, it's just there.

**Leo:** It just shows up, nice.

**Steve:** And I did - okay. So I was also curious to see whether the authentication change would have a retroactive effect, so I went back a week in my Gmail to last week's originally sent email. I have my Gmail account subscribed to this podcast, to the mailings, for exactly this sort of testing. Interestingly, the new GRC logo was NOT also shown on that piece of older email, which I thought was interesting since the email itself does not carry any hint of whether the mailing domain may have a certified BIMI logo. So it appears that Google is checking for the BIMI record at the same time as it's verifying the mailing site's SPF, DKIM, and DMARC status, you know, validating that. And once that's done, and the email has been received, the logo is either established, or it won't ever be.

**Leo:** Oh, hey, I see it. Oh, that's so cool. I just sent myself an email from - is that it, right here over on the left?

**Steve:** Yup, that's it.

**Leo:** That's really cool.

**Steve:** And it used to just show the silhouette of a person, you know, their head and shoulders.

**Leo:** Right.

**Steve:** And now it's actually GRC's logo.

**Leo:** Nice. Very cool.

**Steve:** It is nice.

**Leo:** That's - so you don't have to insert that or anything, it's just it'll always be there from now on.

**Steve:** Yeah, it is based - so it's associated with a domain. And so a query to text records, or that specific text record at GRC.com, returns two URLs which - so one, the SVG, is the graphic. And then there is a signed certificate, DigiCert signed a certificate for that graphic. Both of those URLs get pulled, and that allows the graphic to be affirmatively, like, associated with the GRC.com domain and any email that it generates.

**Leo:** So, yeah, I mean, I don't - I'll have to see if my Fastmail does that. But that's - Gmail definitely does. That's very cool. Very cool.

**Steve:** Yeah. And Leo, break time.

**Leo:** Yes.

**Steve:** And we're going to plow into some feedback from our listeners.

**Leo:** All right.

**Steve:** Actually I've got a neat conversation about some subtleties of third-party or second-factor authentication use. So a goodie.

**Leo:** Good. All right. Oh, and Steve, I finally figured out, I was misusing the instructions for that unlocker. Once I figured out the instructions, I had to download a CMD file.

**Steve:** Ah, okay.

**Leo:** And run that. So that was the automated thing was to download it, and it for some reason couldn't get online or whatever. Oh, I know why. Because Edge said, oh, no, this is not safe. And I'm sure that that's also in the system. Oh, no, we're not going to let you download from something called "massgrave." No, no. You're not going to download that. So once I said, no, no, it's fine, I'm going to download that, I was able to download it, double-click it, gave me the options, worked.

**Steve:** Huh.

**Leo:** Worked.

**Steve:** Woohoo.

**Leo:** Paul thinks, and I think he's right, that actually it emulates an enterprise credential activation site, what's it called, a KWM server. It emulates that.

**Steve:** Yes. That's one of the two activation methods.

**Leo:** That's the loophole; right? Oh, yeah, I'm an enterprise, and I just connect to my server here. You're okay. You're good. It even said "Registered to Leo Laporte." It did the whole thing. So thank you. And I'm not cheating. I bought a license.

**Steve:** Yes.

**Leo:** I just wanted to move that license over.

**Steve:** Yes. You own a license.

**Leo:** And you know why, because the original parallels version of Windows was 22H2. And in order to get 24H2 I actually had to download it from Microsoft and install it. And so that was not activated, so now it is. Thank you, Steve.

**Steve:** Cool.

**Leo:** Thank you, hackers.

**Steve:** Again, you can imagine, you can see how it would be like a preoccupation; right? Like we're going to crack this code.

**Leo:** Oh, yeah. You can also see how it's a really dumb thing to do, what I just did, which is download and run a file from the Internet. But, you know.

**Steve:** You were doing it into a VM, and you...

**Leo:** That's true, it is in a VM, yeah.

**Steve:** Yeah. Okay. So Jaime Denizard, and he gave me his pronunciation, said: "Steve, I've been using Google Authenticator with cloud backup disabled for years, but I would

like to use a more featureful solution, and one preferably not run by Google. The main feature I'm looking for is a solution that has a web portal so that I can get TOTP's from any browser instead of needing my phone with me at all times. How much security would I be giving up, if any, if I went with a solution that offered this such as Bitwarden Authenticator, Ente Auth, or Twilio Authy? Thank you, and keep up the great work. Jaime."

Okay. So I chose Jaime's note because this a question many people have. I get it, like, all - and we've talked about it, but I figured I'd just give it a little more attention, and in the future we'll just refer to this. You know, they want the added security of a second factor, but they don't want the added inconvenience. We've talked about the inherent danger of merging all authentication into a single source, for example, of having one's password manager also supplying the one-time passcode's second factor. Is it as secure as maintaining an entirely separate second factor authenticator and then transcribing the six-digit code manually? No. Is it more secure than not bothering with any second factor? Yeah, of course, absolutely. It all boils down to security models and asking the question, "What exactly are we wishing to protect against?" We need to ask that question because, unfortunately, there are many different points of potential vulnerability.

Okay. So let's address three cases: A full breach of the site being authenticated to, a breach of only the site's known usernames and passwords, or a breach of a user's computer. In the first case of a full breach of the site being authenticated to, the only form of authentication that remains safe after such a full-site breach is Passkeys. Passkeys remains safe because, being a public key authentication system, as I used to say of SQRL, but I'll now say of Passkeys, "Passkeys gives sites no secrets to keep." The only thing a site can do with the public key it has received from its user is verify their identity. It cannot be used in any way to assert or spoof their identity.

One-time passcodes will not protect their users after a full-site breach because one-time passcodes rely upon a shared secret. It's that secret which determines which six-digit code is correct every 30 seconds. So if bad guys are able to obtain the usernames, the password hashes, and the shared secret one-time password seeds, they'll be able to impersonate the site's users. And even if the site is storing its users' passwords as salted hashes, as any modern site now should, a credential stuffing attack that's backed up by having each account's matching second-factor seed would still be able to succeed.

So to recap: In the event of a full-site breach, traditional second-factor authentication, which relies upon the continued secrecy of a shared secret "seed key," would provide no added protection. So it would not matter whether your own authenticator is storing its secret separately or, for example, in your browser.

Okay. In the second case of only a breach of a site's usernames and hashed passwords - or even without any breach, just guessing usernames which are increasingly email addresses, the bad guys would employ, as I mentioned before, a so-called credential stuffing attack. That's the new fancy name, you know, which we used to call "brute force attacks," although credential stuffing suggests that the stuffer is not just guessing randomly, but is instead working from a list of known possible credentials that had been previously harvested from some other service. And this is where reusing passwords between sites becomes a very bad idea.

However, in this case, since the bad guys would not have obtained any of the site's stored second-factor authentication secrets, the use of a second-factor authenticator would strongly protect the user's account. And again, where the authenticator is running, whether it's in the user's browser or offline in a separate smartphone, would make no difference since the bad guys would have no way of guessing the continually changing six-digit passcode.

Okay. So to recap that, in both of the previous two instances of attacks, a full-site data breach or one of the increasingly common credential stuffing attacks, the location of the user's authenticator has no impact and makes no difference.

This brings us to the third case, a breach at the user's end. This could either be a breach of the user's PC with their web browser and its password manager, or a breach of the user's smartphone which contains their second-factor authentication secrets, if that's what they're using. This is the nightmare scenario where the only protection is the separation that hopefully exists between the first and second authentication secrets.

The presumption is that it's exceedingly difficult for any bad guys to get into either of the user's authentication stores, the first or the second factors, because we never see that happen. Right? We're constantly talking about all manner of horrors on the Internet and with Internet-related technologies. But we never encounter instances where users are having their local password managers breached. If I had some wood handy somewhere I would knock on it, since we don't ever want to be reporting that.

**Leo:** Well, there is the exception of the LastPass breach.

**Steve:** Well, okay. But that wasn't a local breach of the...

**Leo:** No.

**Steve:** That was headquarters being breached.

**Leo:** Right.

**Steve:** Right.

**Leo:** Okay, yeah.

**Steve:** Yeah. So it's not the actual, you know, we don't see, we're not ever reporting stories of, like, some problem with some password manager that turns out has a horrible problem.

**Leo:** Right.

**Steve:** And so this substantiates our intuitive sense that it's safe for us...

**Leo:** Oh, except for RoboForm, which was used to hack people's wallets; right? So...

**Steve:** It was - that was a...

**Leo:** Because it had a non-random...

**Steve:** ...bad number generator.

**Leo:** Yeah, it had a bad random RNG, so...

**Steve:** Yeah.

**Leo:** Yeah. Your point is valid, absolutely.

**Steve:** Right. So the point is all the evidence we have, not only theoretically but practically, is that we're not seeing problems with password managers being able to keep their secrets.

**Leo:** No.

**Steve:** They are. And given that it's exceedingly difficult to break into one credential store, it's beyond exceedingly difficult to imagine that two separate credential stores using wildly differing technologies - a PC and a smartphone - might both be simultaneously compromised in order for bad guys to obtain both first- and second-factor secrets and then facilitate spoofing authentication.

Okay. In other words, the only danger posed by storing both the first and second authentication factor secrets in the same place, in the same device, and thus under the same form of protection, is that the security of that device could possibly, conceivably, be breached. And moreover, we're aware of no instances where that has happened or has been a problem.

**Leo:** So the MFA is not stored in a vault. The secret is stored in the vault on LastPass's servers, though, or Bitwarden's servers.

**Steve:** Actually, copies are downloaded to your local browser.

**Leo:** Yeah, but I'm just saying, if those sites, if as what happened with LastPass, if the vault has been exfiltrated...

**Steve:** Ah, I mean, that's a good point. If they...

**Leo:** Your secret is in that vault.

**Steve:** If they're holding both first and second, and headquarters is breached, then like all their users are up the creek.

**Leo:** I mean, this is highly theoretical.

**Steve:** Right.

**Leo:** And as we said before, you know, you're probably fine doing this.

**Steve:** Yes.

**Leo:** I have a separate app just for that reason, that's all.

**Steve:** And actually I wrote in the show notes, so at this point today, it's only a theoretical concern and argument. But it is nevertheless a concern and an argument, no matter how theoretical it may be. Which, you know, we've just brought up; right? It could happen, and something did happen at LastPass.

So, you know, this is very much like our recent discussions of whether it's safe to leave an otherwise unprotected Wireguard VPN service port exposed and listening on the Internet as tens, if not hundreds of thousands, of people do. As I said last week, it's very much almost certainly safe. There's every reason to believe that it is safe, and no reason to believe that it isn't, right up until the moment that we learn that it wasn't.

**Leo:** Right.

**Steve:** So, you know, I'm spending so much time on all this because it's an important concept that binds these together. The concept is "layered security." The idea of layered security is that no single fault, vulnerability, or compromise in the security of something protecting a system would result in a compromise of that system's security. Another more colloquial term for "layered security" would be "belt and suspenders." I would always put WireGuard - I would - behind some other form of access control, if only so that any failure of either one would not result in a failure of the whole.

And the concept of "layered security" is what gave us multifactor authentication in the first place, not relying upon any single factor. If one is compromised, the other can be trusted to hold. Ideally, the implementation of layered security doesn't pose an ongoing burden upon its user. And this is where the implementation of the system comes into play. If the machine a user is authenticating from already contains a reasonably fresh previous authentication cookie, depending upon the security needs of the website, it would be reasonable to bypass the request for the user's second factor and only ask for it if either a long time has passed since the user last authenticated from that machine, or if the user is authenticating from a machine that has no record of previous authentication.

This model continues, you know, the model of doing that, only prompting for second factor when there's some reason to do so, you know, still strongly protects the user from an online credential stuffing attacker, for example, whose authentication guesses would not carry the second-factor bypass cookie, while also reducing the annoyance factor to repeat users of the same machine.

So, Jaime, your question was obviously a good one because it certainly didn't have a short answer. And the answer that it did have is best viewed in the context of the various possible threats that it needs to protect against. Practically speaking, I think a good case could be made for most users to just let their existing password managers painlessly supply their second-factor one-time passcodes for them. That provides strong protection against the known online password stuffing style attacks that we know are occurring, and

against those attacks it is providing layered belt and suspenders authentication protection. The fact that it is not also protecting from a theoretical attack that we have no evidence of ever having happened, you know, not being a problem, even though the protection could be provided by moving those second-factor secrets to a different device, is almost certainly taking caution too far, until it isn't.

I keep my second-factor tokens in my smartphone. My browser doesn't have them. They're not online, except in the sense that they are synchronized through iCloud and stored encrypted for the sake of synchronizing, and I appreciate that. Although I really don't have to do that, either, because I add them so infrequently, and I always print out the QR code if I need to synchronize devices or, you know, to restore.

**Leo:** Well, there's your weakest link. If somebody breaks into your house, he's got the QR codes. Now you're really in trouble.

**Steve:** There are not any Russians or suspicious-looking foreigners lurking around, so...

**Leo:** I think that's really the other side of that equation is how much harder is it to store it in a separate program. I don't consider that a big jump in difficulty, so I do it.

**Steve:** Yes. Jaime is saying his bar is lower.

**Leo:** Right.

**Steve:** You know, it bugs him having it - and maybe he's using some sites that are not well designed, and so they're constantly asking him when he's sitting at the same non-shared PC, it's like, I just gave this to you yesterday.

**Leo:** Yeah. That would be annoying, yes.

**Steve:** Yeah. I mean, most of us have static IPs, so the site could encrypt our IP into the cookie so that it can see if the, you know, I mean, there are all these things that could be done where, you know, like properly, to properly use a second factor. And it's unfortunate when sites don't, you know, bother.

**Leo:** More and more I'm seeing sites forcing me to reauthenticate a lot.

**Steve:** Yes.

**Leo:** And it's really annoying.

**Steve:** Yes.

**Leo:** But I guess we live in a dangerous world.

**Steve:** Nir Eden said: "Dear Steve, I've been a dedicated listener of Security Now! for many years. Your show has expanded my technical understanding and reinforced important values I deeply believe in, particularly that privacy is a fundamental condition for freedom, accountability to the entire Internet community, and unwavering reliability. Regarding remote access solutions: While overlay networks like WireGuard and Nebula work well, they lack granular access control and can be complex to set up. Solutions like Cloudflare tunnel and Ngrok provide public-facing interfaces, but I needed something different. I wanted to create a private tunnel from my home Raspberry Pi SSH server to my laptop so I could log in from anywhere. I wanted to connect a cloud web server to a micro-service that runs on another cloud. I wanted to link database servers and clients running on different locations."

He says: "I developed a solution based on SSH tunneling through an external server. Since both ends make outgoing connections, opening ports or modifying firewall settings is unnecessary. I have developed a simple web interface, so connecting two devices is as simple as setting up a Zoom meeting," meaning clicking on a link. "After using it successfully for years to connect cloud services and remote control devices, I've made it publicly available at [www.puppetpc.com](http://www.puppetpc.com)." He says: "It is currently free to use, as I want to see how far this solution can go. Thank you. Nir Eden."

I went over to [www.puppetpc.com](http://www.puppetpc.com) and took a look around. The site looks very clean and new, and I imagine it will evolve over time. There is not yet any deep technical documentation that I could see. So I know that many of our listeners would need to know why they should trust it. But I'm aware that others won't care that much and may just be content to play with whatever it is. So I'm not vouching for it in any way, since I cannot. But I wanted to share this very nice-looking creation of one of our listeners, to give Nir some attention to his efforts that might be useful to him, and to reiterate how amazed I am by the quality of the people who choose to spend their time listening to this podcast. So thank you for the share, Nir: [www.puppetpc.com](http://www.puppetpc.com).

Steven Cedrone wrote: "Hi, Steve. I heard you mention Tor's call for more bridge operators in SN-1003," last week. He said: "I wanted to bring to your attention the Snowflake extension/add-on for Firefox, Chrome, Brave, or other Chrome-based browsers. It allows the Tor Network to use your computer as a proxy to help people circumvent censorship, and it's as easy as installing a web browser extension/add-on. You can also toggle the settings to allow it to continue running, even when the browser is not open. They're good about not slowing down your Internet connection, and they hide your IP address while someone is connected through your computer. The Snowflake also changes from purple to green in color, if pinned to the toolbar at the top, so you know when someone is currently connected."

He said: "I want to mention this to you in hopes people might help the Tor network in this way, as well, because not everyone has the skill to run a server to run a bridge as I do. Not the easiest to set up in Linux," he notes. He says: "Read more about it here," and then he gives the URL [snowflake.torproject.org](http://snowflake.torproject.org). Okay. So that is very cool. I love that something like this could so easy to set up and be safe to use. The Tor Project folks certainly know what they're doing.

And just to explain, this Snowflake, this proxy, serves as a middleman in between nodes. The Tor servers do all of the fronting of connections. But as we know, it's very useful to bounce traffic around a while within the Tor network in order to increase its security. So that's the - so you're not an end node. Nobody sees your IP address. You're one of the internal nodes that just gets used to scramble the traffic up. That's how Tor is able to keep from overloading your bandwidth.

One of the reasons I'm very glad Steven put this on our radar is that these days most of us have massive bandwidth overkill, with our bandwidth mostly sitting idle. So the idea that we might be able to donate some small piece of our bandwidth to help the Tor Project and to provide some more diffusion seems like a great idea. I followed Steven's link and went over to the Tor Project's Snowflake page. It turns out that Snowflake's function as a traffic proxy is only one of the things it's able to do. It also allows the users who install it to use the Tor system.

So they said: "Snowflake is a system that allows people from all over the world to access censored websites and applications. Similar to how VPNs assist users in getting around Internet censorship, Snowflake helps you avoid being noticed by Internet censors by making your Internet activity appear as though you're using the Internet for a regular video or voice call. There are numerous tools available, such as Snowflake, that 'transform' Internet activity, each using a different technique." And they mean numerous Tor tools.

He said: "Some redirect Internet traffic to appear to be coming from popular cloud providers like Microsoft Azure and Amazon Web Services. Others scramble Internet traffic in order to make it appear completely random. It therefore becomes costly for censors to consider blocking such circumvention tools since it would require blocking large parts of the Internet in order to achieve the initial targeted goal.

"Unlike VPNs, you do not need to install a separate application to connect to a Snowflake proxy and bypass censorship. It is usually a circumvention feature embedded within existing apps. Currently Snowflake is available inside Tor Browser on Desktop and Android, Onion Browser on iOS, and Orbot on Android and iOS. If you've downloaded and installed any of these apps, and they are censored in your country, you can bypass the censorship by activating Snowflake through the app's settings page."

And then we get to the part that caused Steven to write his note. The Tor Project writes: "Did you know that Snowflake proxies are operated entirely by volunteers? In other words, a Tor user gets matched with a random Snowflake volunteer proxy, which is run by a volunteer like you. So if you want to help people bypass censorship, consider installing and running a Snowflake proxy. The only prerequisite is that the Internet in your country is not heavily censored already. You can join thousands of volunteers from around the world who have a Snowflake proxy installed and running. There's no need to worry about which websites people are accessing through your Snowflake proxy. Their visible browsing IP address will match their Tor exit node, not yours. There are various different ways to run a Snowflake proxy, beginner to advanced."

And then it said: "Install the web extension. The web extension is the easiest way to run a Snowflake proxy. Simply install it on Firefox, Chrome, or Edge, enable the extension, and watch the icon turn green when a user connects through your proxy."

**Leo:** Oh, that's cool. I have installed it on my browser, and it's up in this upper right-hand corner. It's very small. It's purple right now. But that's cool. I'll know when somebody's using it. It'll turn green. Oh, that's neat.

**Steve:** And you're just - you have become part of the Tor network while you choose to have your browser open. Or even if you select an option, you'll allow it to keep running even if your browser's closed, but as long as your computer is on, obviously. And it allows you to be part of the mixing of traffic that the Tor system is providing.

**Leo:** I think that's where they [crosstalk], especially nowadays, yeah. I think we need it now.

**Steve:** Yup, very cool. So thank you for bringing the Snowflake to our attention, Steven.

John Robinette has a solution for linking smartphones and PCs. He said: "Hey, Steve. With regard to your wish for a way to easily type something on your PC and send it to your iPhone, I would recommend LocalSend." And he referred to [localsend.org](https://localsend.org). He said: "The simplest way to describe it is a cross-platform AirDrop, written in Dart+Flutter, that works on iPhone, Android, Linux, Windows, and Mac. It does require installing an app, but the communication is all local between devices. LocalSend uses mDNS to discover other LocalSend clients on your subnet, which then allows you to send and receive text, files, photos, and so on." He says: "I've been using it for about a year to move various files between my Windows PC, iPhone, iPad, and a Linux PC." So very cross-platform.

He said: "If you don't want to install an app, there's also PairDrop at [pairdrop.net](https://pairdrop.net), which is similar, but entirely browser based. The actual transfer of data is peer-to-peer via WebRTC. However, establishing this peer-to-peer connection depends on both clients first making a connection to the website, so it won't work if your Internet connection is down, or if you're paranoid about using someone else's server. But it's open source and easily self-hosted if you're that person. Hope that one of those or both of those might be useful for you or others.

**Leo:** And of course nowadays, on a Mac anyway, you have this iPhone access, so you can use this on your Mac.

**Steve:** Right.

**Leo:** And on PCs you have it for Android devices. Actually you can sort of use it with iPhones, as well.

**Steve:** And apparently that is the case. And actually we're about to get to that.

**Leo:** Oh, okay, sorry.

**Steve:** Oh, no...

**Leo:** I like these two apps, though. That's really nice, yeah.

**Steve:** Yes, they are very - and very, very cross-platform. Jay Soch said: "Good afternoon, Mr. Gibson." He's being formal.

**Leo:** Mr. Soch, pleased to meet you.

**Steve:** He says: "Long time, first time. You got me into Bug Bounty, and I now make a not insignificant amount of income through Bug Bounty."

**Leo:** Ah, interesting, huh.

**Steve:** Yeah. He said: "My wife is obsessed with" - oh, this is the SodaStream guy. "My wife is obsessed with La Croix, and we've spent a lot of money on it over the years." He says: "This year I'm thinking about getting her a SodaStream-like device" - 'tis the season - "so she can get her fix more easily, and we can hopefully save some money. I remember that you discussed some techniques you had used to save some money on a similar device on the podcast, and I am going to go through the notes and find that information.

"What I would like to know is if you have any updates to your previous process? As I recall, you had changed the adapter on the CO2 cartridge and were getting your CO2 canisters refilled somewhere in Irvine." He said: "I'm in Fullerton. Do you still do this? Has this process held up over many uses and years? I would love any thoughts you have on whether this is a worthwhile investment or not. Thanks for sharing your very valuable time."

Okay. I'll take up just a bit of everyone's valuable time because it has been such a win for us. The trick is to have a single large CO2 master tank that's used to directly refill empty SodaStream canister little mini tanks that the SodaStream uses. This allows you to perform the refilling from the big tank to the little tank at home, using the SodaStream canisters over and over again. And really part of it was saving money. It was just annoying to have to, like...

**Leo:** Yeah, it's a pain to ship those back.

**Steve:** ...continually recycle these canisters, yes. And the master tank can, in turn, be filled over and over by any home brewing shop.

**Leo:** Now, I was - I tried this, and I was unable to find anybody who was willing to do that.

**Steve:** Oh, okay. So...

**Leo:** You may have lucked out there in Irvine.

**Steve:** Yeah, I've got one off of Bristol, like a mile away. And so people - so you do want to verify that first; right?

**Leo:** And you may want to get the tank from them because that was one of the issues is a lot of people said, well, I'm not going to fill some strange tank. They wanted to know...

**Steve:** Okay.

**Leo:** ...it was something that they had...

**Steve:** And I imagine that the tank from them is probably no more expensive.

**Leo:** Right.

**Steve:** Although it does have to be a special tank. So first of all, people who brew their own beer at home use the same tanks and get them refilled. Okay. The first trick is interconnecting the two tanks, and Amazon has plenty of adapters for exactly this purpose. They are typically nicely machined brass adapters that have a valve. One end of the adapter fits the empty SodaStream canisters, and the other end mates with a standard CO2 tank which is also available from Amazon. I believe mine is a 20-pound tank.

**Leo:** I bought one and then palmed it off on Mikah, and he didn't want it because he couldn't get it filled. So I think we gave it away.

**Steve:** Yeah. So...

**Leo:** I wish I had known, I would have sent it to our correspondent.

**Steve:** Yeah, definitely make sure you're able to fill it. They're about \$150, so they're not inexpensive.

**Leo:** They're not cheap, yeah.

**Steve:** But it's light, the 20-pound one was light enough for me to drag and roll from my car to the shop for refilling and back. The only requirement for the tank is that you need to be sure to get one with a so-called "siphon tube." The siphon tube extends from the valve on the top all the way down to the bottom, and that's what allows you to fill the empty SodaStream canisters with liquid carbon dioxide taken from the bottom of the tank, rather than CO2 gas which would be taken from the top. And as I said, they're not inexpensive. They're about \$150. But it's been worth it for us. And I have not counted the number of times we're able to refill a small canister from the much larger tank, but it's many, many, many times. I mean, I think I've only gone to the home brewing place maybe three times in total. And they had no problem refilling the tank.

**Leo:** Okay.

**Steve:** But you might, you know...

**Leo:** Check first.

**Steve:** Certainly it makes sense to buy it from them, as long as they can provide you with one with a siphon tube.

---

**Leo:** Right.

**Steve:** Because you do need to have that for sure. Otherwise you have to turn the thing upside down while you're doing it.

**Leo:** That's no fun.

**Steve:** You don't want to do that. Okay. Finally, Troy in Montana suggests Intel's Unison. But then there was one other. He said: "Steve, long-time listener since day one. If you have an Intel PC" - well, we know I have those - "you can use their app to connect your iPhone to a PC and get access to sending messages. Not perfect, but a way to do what you hoped." And then he provides a link to Intel.com, and it's something that they call Unison. He says: "Thanks for all you and Leo do to keep us safe."

**Leo:** So, okay. I was excited when I read Intel's description. It says: "Following a simple pairing process between the phone and the PC, you can make or take phone calls from your PC, send or receive text messages using the PC's mouse and keyboard, and view phone notifications on the PC screen. Also, you can seamlessly and bidirectionally share photos, videos, and documents between your phone and your PC. The Intel Unison solution fully supports both Android and iOS."

And, I mean, like their - I put a picture of what they have on their website in the show notes. It looks like a full-size desktop version of iMessage on the screen with all the contacts and messages shown. I mean, it looks utterly amazing. But when I drilled down a bit more, I tripped over the following: "The Intel Unison application is available for download on any Windows 11 PC that meets the minimum requirements, as detailed in the app store descriptions. Both laptops and desktops are supported."

So for anyone who has already made the move - what was it the Microsoft guy said, who had "transitioned" to Windows 11, yes, that's right - it really looks like more than I could have ever dreamed of. You could actually have, apparently, a functioning messages app sitting on your Windows desktop with your phone nearby on its charger. I already have a need to run a Windows 11 VM since the work on the DNS Benchmark, which is what I'm doing now, has turned up some subtle but important differences in Windows 11 handling of some app resizing. So I was planning to get Windows 11 set up under VirtualBox in any event. But if I could load this onto that machine, I might have Windows 11 VM running 24/7. So thank you for that, Troy.

And then Henrik Johnson said: "Unfortunately not available for Windows 7, but supported in Windows 10 and later." He said: "You can check out Phone Link. Should be built into Windows and enabled by default." And there's a link in the show notes, and the URL ends with "sync-across-your-devices." And Henrik says: "You can read and answer texts, see any notifications, and even make phone calls from your computer. It uses Bluetooth underneath to make the magic work." He says: "I just switched from Android and was really missing Pushbullet, but this is a pretty solid replacement."

Okay. So it appears that my prayers may have been answered, and that the frustration I've been feeling has not been mine alone, and that solutions to this have been created. I don't know how or whether this is related to Intel's Unison, but it

looks like the same thing, and Intel is just sort of private labeling the same Windows application.

I was a bit nervous because I tracked down Phone Link, and Microsoft says requires Windows 11. But Henrik clearly said Windows 10, so I'm hopeful. Microsoft might just be refusing to in any way promote the continued use of Windows 10, so like they've just scrubbed it from their website. So anyway, thank you, Henrik, and also thanks to all of our listeners who heeded my call and my pleas for a solution. And Leo?

**Leo:** Yes.

**Steve:** After this break, let's talk about GPT.

**Leo:** All right. And by the way, I've been using Phone Link for some time on Windows 11. I'm not sure if it works on Windows 10.

**Steve:** Okay. So tell me.

**Leo:** But I bet it does. It works best with Android, but you see we can send text messages. I'm connected right now to an Android device, my Z Flip, my Samsung Z Flip.

**Steve:** Okay.

**Leo:** And it really does work best with Samsung. You know, but I've been able to use it with iPhone. It just doesn't have all of the features.

**Steve:** I don't need all. I just need my sanity preserved.

**Leo:** You want text, yeah. Well, I can't promise you that, Steve. You know that.

**Steve:** Oh. That may be a, yeah, that may be a - yeah.

**Leo:** All right. Let's get back to Security Now!.

**Steve:** Okay. So I'm going to warn our listeners that the introduction here is dense. But it is not actually important to understand every nuance of what I'm going to explain, although some people will find it interesting. So for them, and because this is what happened, I want to, you know, in order to set this up for the conversation that I have, it's important.

Okay. So as I mentioned at the top, I had an interesting interaction with the coding version of ChatGPT 4o, which they call "4o with canvas." And that's while I was working

on the update that I'm working on to GRC's DNS Benchmark. And as I've mentioned recently, I've been using the coding platform version as sort of a super Google search on steroids. I'm often astonished by the quality of its replies. Something that I don't understand happened over the weekend while I was working on code. But frankly, I don't understand any of this AI stuff. It's all voodoo. And that's the problem because I'm 100% certain that this is too important for us to not understand. And I have a plan for that, but let me first share what happened.

Okay. One of the facilities of Microsoft's Macro Assembler which I use to make my assembly code more concise, that is, one of the features that I use, and more legible, is the assembler's macro facility. I have a macro named, happens to be named "AppendRichEdit," which takes a string argument. That is, the macro takes a string argument. So in my program code I would write, for example, AppendRichEdit, and then in quotes "Benchmark Results." Now, the way assembly language macros work is that when the source code is being assembled, the assembler does what's called "macro expansion," which causes it to follow the simple macro script to create additional code from that script. The point is that this is all nicely hidden behind the macro, which just says AppendRichEdit and then a string in quotes, which makes for a more readable program.

In the case of my AppendRichEdit macro, which as I said takes a string argument, when the code is being assembled, the macro script places that string argument into the program's data section, then it writes a call to my AppendRichEdit function, passing it a pointer to that string. I could have done the same thing by hand, but this creates a much clearer communication. And one thing I've learned from, yes, 55 years of programming, is that coding is all about communication. Almost as much about communicating to me as it is to the computer, which is why my code is, frankly, it's beautiful. I mean, the computer doesn't need it. I've seen people write assembly language where there's a bunch of opcodes down the left-hand margin of the page, and it's like, what is this crap? Mine, you know, it's about communication.

Okay. So the use of this macro simulates the semantics, which everyone is used to in high-level languages, where it's possible to use a literal string as an argument in a function call. This would be like writing in the BASIC programming language, you know, you would write Print "Hi Mom!" It's very convenient. But in assembly language, the string "Hi Mom!" needs to be defined elsewhere in a data section of the program, and then the address of that string is provided to the Print function for it to print. So this is very efficient if you might have some repeated use of the string "Hi Mom!" throughout the program, since all of those repeated instances can all reference that single "Hi Mom!" data string.

But the need to define the string elsewhere in the program, that is, from where you're using it, makes the resulting code somewhat less clear. By default, assembly language doesn't offer the high-level language convenience of in-place string declaration and use, so I use a macro to give me the same semantic flexibility. Essentially it looks like a higher level language is being used, although it's still low level underneath.

Okay. So I apologize for the long and esoteric, you know, "inside baseball" explanation, but I wanted to explain the situation surrounding what I was about to ask ChatGPT 4o about. What I needed was the ability to optionally add another argument to the macro. If that optional argument was present, it would be provided to the function call which the macro wrote for me. And if I did not supply that optional argument to the macro, the macro would provide a default argument in its place.

Now, this is not a feature of MASM. MASM is Microsoft's Macro Assembler, M-A-S-M, that I frequently use. But I knew that MASM allows the specification of both required and optional parameters. What I wasn't sure about was having it supply default values for

non-specified parameters. But since MASM has a very powerful macro facility, thus it's called a Microsoft Macro Assembler, I had the sense that it probably had the capability to supply a default value for missing optional macro parameters.

To get the details of something like this, once upon a time, back in the old days, which in this case is about two months ago, before I started using ChatGPT for these things, I would have googled around a bit to see whether I could quickly find an answer. But more often than not, since the sorts of things I'm looking for tend to be somewhat obscure, I would wind up reading a couple of pages of the MASM documentation. But now we have ChatGPT, which, as I've mentioned, I've been using more and more. So the first thing I do now is ask it.

So I got a ChatGPT prompt, and I wrote: "Can I use a macro in MASM where an optional macro parameter has a default value if it's not specified?" I mean, that's exactly the phrase I put in. To which ChatGPT 4o quickly replied: "Yes, in MASM" - and it helpfully added (Microsoft Macro Assembler) - "you can create a macro with optional parameters that have default values if they are not specified. This is achieved by using the IFB directive to check if a parameter is blank." And it says, "(i.e., not provided) and then assigning it a default value within the macro. Here is an example to demonstrate how to define and use such a macro." And then it gave me some sample code.

Okay. Now, right off the bat, the syntax it showed for the macro seemed suspicious. The way it was specifying a default value looked wrong to me. It was a construction that various higher level languages use. But I thought, well, I did ask. So I modified my original macro definition as ChatGPT had specified, and MASM was not at all pleased with it.

Since ChatGPT seemed quite clear and authoritative in its answer, I first made sure I hadn't made some mistake, but everything looked good. I tried tweaking it myself, and doing some various things, sort of variations on that theme, but nothing I tried worked. So I replied to ChatGPT, and this may be the first time I've ever done this. I wrote: "That didn't work. The syntax of the first line of the macro definition was incorrect. Any ideas?" And ChatGPT replied: "You're right."

**Leo:** You're right. I'm an idiot.

**Steve:** It's like, yeah. It said, "You're right."

**Leo:** Yeah.

**Steve:** "MASM's macro syntax doesn't directly allow you to specify default values inline in the macro definition." And then it quoted its own previous incorrect sample code with an example. And for example, the quote is Param2=<DefaultValue>. Then it said: "However, you can achieve the same effect using the IFB directive to test for blank parameters and handle them manually. Here's a corrected example." And then it gave that. And although I needed to modify what it provided for my needs, that second attempt worked.

Okay. So to recap, I asked this thing, which often conversationally refers to itself in the first person, which is still a little annoying and jarring, it's an it, not a, you know, it doesn't have pronouns. So I asked this thing a question. It quickly provided an authoritative-sounding and reasonable answer which was nevertheless wrong. Then I

pointed out the part of the answer that was wrong, and it said, "You're right," and then proceeded to offer a corrected answer.

But I remained haunted by my original impression that MASM did in fact allow default values to be specified for optional and missing parameters. You know, it's kind of the way my mind works. I still remember all of the WordStar keystrokes because they were, you know, built into me, and they've got some deep neurons in my brain because I used them so much. So I went to the source, which is where I would have wound up in the old days. And the MASM Reference revealed the truth. It is indeed possible to specify a default parameter for an optional and unspecified macro parameter. ChatGPT was almost correct the first time. It simply forgot that a colon and an equals sign were needed to separate the parameter and its default value, instead of just an equals sign.

So it gave me an answer that was initially wrong, but was almost right. When I told it that there was a problem with that answer, it agreed, saying, "You're right, MASM's macro syntax doesn't directly allow you to specify default values inline in the macro definition," which that statement later turned out to be wrong. And it gave me what was effectively a workaround for the feature that it then assumed was missing from MASM.

Finally, having found the correct answer to this, I decided to let it know. So I replied: "Following up on this, it turns out that your" - I'm saying "your." Okay, I don't know how else to address it. "Following up on this, it turns out that your first response was almost correct. The proper syntax for the first line of the macro definition is: 'MYMACRO MACRO Param1, Param2:=<DefaultValue>.'" I said: "The thing that was missing was that colon (:) in front of the equals sign." Okay, and I hit ENTER.

And the damn thing responded: "Ah, you're absolutely correct. Thank you for pointing that out. In MASM, the syntax for specifying a default value for a parameter does indeed require the := operator, not just =. Here's how it should look," and it provided a correct example. Then it provided me a little tutorial with the heading "Why := Works," writing, it wrote to me, "The colon in := is required to differentiate default values from other constructs in MASM. Without the colon, MASM doesn't interpret the equals sign as part of the default value definition. Thanks for catching that nuance! It makes the macro much more concise and straightforward than relying on IFB." And then it actually gave me a smiley face. Okay. So...

**Leo:** I've got bad news for you, though, Steve, unfortunately. It's not going to remember that. So if you ask it again, it's going to still make the same mistake.

**Steve:** Oh, and thank god.

**Leo:** Because unfortunately it doesn't. You're not allowed to change the training.

**Steve:** Because can you imagine the garbage...

**Leo:** Exactly.

**Steve:** ...that the Internet would be filling with.

**Leo:** It might remember it locally because it does remember some stuff locally. But you can't teach it because obviously, if you could, it'd be a problem.

**Steve:** Oh. The world, the end of the world as we know it.

**Leo:** It was very polite about the correction. That is a very useful prompts tip that a lot of people have noted is you can say, no, that's wrong. And it will actually come back to you and often get it right. So it's very interesting. No, I'm not dead yet, something like that.

**Steve:** I wanted to share this conversational event because I'm still startled...

**Leo:** It's amazing.

**Steve:** ...by this thing. And you heard what it apparently said about our Picture of the Week.

**Leo:** Yeah.

**Steve:** It was just, like, amazing.

**Leo:** Yeah, very impressive, yeah.

**Steve:** You know? And because I was left staring at the screen wondering "What have we created?" The fact that I really have no idea is unnerving. And I know I'm not alone in being unnerved by this. Whatever this is, as I said several weeks ago, I believe it's the biggest and most significant transformative event of our lifetimes. Aliens have not landed in our backyard. We have created them.

**Leo:** You know, it's funny you should say that because I have a friend who works in the business. And he said that's a better way to think of it is as an alien intelligence. It's just different from ours.

**Steve:** Yeah.

**Leo:** Yeah.

**Steve:** And it seems clear that this is just the tip of the iceberg. Now, I have a picture at the end of the show notes, Leo. I have spent my entire life...

**Leo:** This is amazing, by the way. I love this picture.

**Steve:** ...working to understand the way things work. And I have proof of that. The last page of the show notes has a photo my dad took of me at age four...

**Leo:** Four.

**Steve:** ...at the picnic table in our backyard in Orinda, California.

**Leo:** Wow.

**Steve:** I needed to understand exactly how electricity worked and why you couldn't just hook up one wire to a light bulb.

**Leo:** Oh.

**Steve:** Because why not; right? The electricity comes out of the battery and goes to the light bulb.

**Leo:** Right, right.

**Steve:** Nothing has changed since then.

**Leo:** Did your dad make you that board?

**Steve:** No. No.

**Leo:** You made the whole thing?

**Steve:** Yeah. I mean...

**Leo:** Oh, my god.

**Steve:** ...you can see the way the kite string is wrapped around the dry cell battery.

**Leo:** Yes, yes.

**Steve:** A little excessively.

**Leo:** Hey, it's not going anywhere. Wow.

**Steve:** So I wanted to understand this. You know, nothing has changed since, you know, me at age four back then. Today, I want to understand this, whatever this is. So two days ago I identified and purchased two quite lengthy, technical, and detailed textbooks on the subject of Large Language Models, Conversational and Generative AI.

I am blessedly, and finally, nearing the end of Part One of Peter Hamilton's seemingly endless two-part Archimedes Engine novel series. Once I'm finished with that, I'm going to turn my attention to educating myself about AI, and not just for myself. I have every intention and expectation that I'm going to reprise my role as Security Now!'s Explainer in Chief to explain to this podcast audience exactly what I've learned about what we are creating. I need to know, and I'm pretty certain that among this audience I'm not alone.

**Leo:** You are not alone.

**Steve:** So stay tuned!

**Leo:** Can't wait, Mr. Explainer in Chief. Yeah, it's a fascinating subject.

**Steve:** It's just mindboggling, Leo. I have no idea. I understand how all this other stuff works. This, I just don't have a clue.

**Leo:** Well, to some degree it's a black box.

**Steve:** [Crosstalk] there are people who do.

**Leo:** I mean, you can understand how it's trained, and you can understand roughly how it works. There's a very - I recommend, for a shorter version of these longer books, Stephen Wolfram has done a really excellent explainer of how they work, as one would expect. He's done a lot of writing now about AI. He's very interested. But the problem is the rules they generate are not visible and are essentially a black box. And so that's kind of an interesting - I see you looking it up right now. That's great.

**Steve:** Well, yeah, I didn't want to lose that. We certainly know that's the case with neural networks; right? You know, they adjust their strengths based on being trained.

**Leo:** Right.

**Steve:** And adjusting their outputs to match what is told they should be. But we don't actually understand the weightings...

**Leo:** What's in there. That's right.

**Steve:** ...of the neural network. It's just what it does, and it works.

---

**Leo:** Transformers are basically a form of neural network. So it's very similar. I will be very interested to see what you can figure out. I can't wait.

**Steve:** Well, I intend to do a Security Now!-style explanation of this, once I understand it myself. So we'll see what we get. I don't know what's going to happen.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the Creative Commons License v2.5. See the following Web page for details:  
<http://creativecommons.org/licenses/by-nc-sa/2.5/>