A Clever Regurgitator

Description: For how long were bad guys inside GoDaddy's networks? What important oral arguments is the U.S. Supreme Court heading today and tomorrow? What's Elon done now? What's Bitwarden's welcome news? What's Meta going to begin charging for? Should we abandon all hope for unattended IoT devices? Are all of our repositories infested with malware? How did last Tuesday's monthly patch fest turn out? Why would anyone sandbox an image? What can you learn from TikTok that upsets Hyundai and KIA? And are there any limits to what ChatGPT can do, if any? We're going to find out by the end of today's 911 emergency podcast.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-911.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-911-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We'll answer the musical question, how long were bad guys inside GoDaddy's network? We've got some good news for our sponsor Bitwarden and its customers. And then he's going to talk about ChatGPT. How useful would ChatGPT be at detecting malware? It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 911, recorded Tuesday, February 21st, 2023: A Clever Regurgitator.

It's time for Security Now!, the show where we get together and talk about security. Right now. Steve Gibson is here. Hello, Steve.

Steve Gibson: You think that's how we came up with the name, Leo? I think that might have been.

Leo: I don't know. I don't know. Well, better than Security Yesterday!.

Steve: Oh, yeah. That's, you know, that's has-been security.

Leo: Nobody cares about that.

Steve: We don't want that.

Leo: Yeah.

Steve: No. So we're here to answer some questions, as we've been doing so far this year. One is how long were bad guys inside GoDaddy's networks? What important oral arguments is the U.S. Supreme Court hearing today and tomorrow? What has Elon done now? What's Bitwarden's welcome news? What's Meta going to begin charging for? Should we abandon all hope for unattended IoT devices? Are all of our repositories infested with malware? How did last Tuesday's monthly patch fest go, anyway? Why would anyone sandbox an image? What can you learn from TikTok that upsets Hyundai and KIA? And are there any limits to what ChatGPT can do, if any? We're going to find out by the end of today's 911 emergency podcast.

Leo: I'm going to give you the short version so you don't have to listen to the whole thing: A long time. Gonzalez v. Google. Tweet. Argon2. Verification. Yes. No. Yes. Yes. How about that?

Steve: Very nice.

Leo: Very good. We will get to the actual answers.

Steve: We'll see you next week.

Leo: If only it were that simple. All right. Picture of the Week.

Steve: So today's Picture of the Week, or this week's Picture of the Week, was actually taken by one of our listeners who was up in the attic of some sort of charitable organization, maybe his church. I don't quite remember now what he said. But this was a - he was working on fixing their Dish Network installation. And when he saw the ground wire attached to a nail that was nailed into some wood, he thought, okay, I've got to take a picture of this and share it with the Security Now! audience because here we have another weak understanding of the goal of grounding.

Leo: Where does the other wire go?

Steve: It's not clear. It wanders off somewhere. And what occurred to me was that maybe whoever it is who installed this thought maybe that the electrons would pay attention to the color of the insulation.

Leo: It's green.

Steve: Because, you know, if they realized that it was a green wire - traditionally in electronics, electricity, you know, green is ground. So they go, oh. Everybody over this way. Of course the problem is when they get over to the nail, which is stuck into some wood, wood is a very good insulator. So it's a little bit like sticking the wire into that pail of dirt, which is one of our all-time favorite pictures. So anyway, thank you very much to

our listener Mark for thinking of us when he thought, what's wrong with this Dish Network installation? Oh.

Leo: Don't you love it? When they see something like this, they think of you immediately; right? Send it to Steve.

Steve: Okay. So I titled this one "GoneDaddy." Last Friday GoDaddy revealed a rather astonishing bit of news. Its network and organization had suffered a multi-year security compromise that had allowed attackers who to this day remain unidentified to exfiltrate the company's source code, customer and employee login credentials, and install their own malware which redirected customers' websites to malicious sites.

Leo: For years.

Steve: Years, yes.

Leo: Years.

Steve: So, you know, they're big; right? They have got nearly 21 million customers.

Leo: They're the number one registrar in the world. They're huge.

Steve: Their last year revenue was nearly $4 billion. So many years ago, when I was making my move away from Network Solutions, I gave GoDaddy some consideration. It is the choice of a very techie friend of mine, whom we both know, Mark Thompson, maybe because he's in Arizona, and I think that's where they're based also. But for me it just looked too bubblegum...

Leo: They're terrible, yeah.

Steve: ...and commercial. You know?

Leo: I'm not surprised to hear this.

Steve: Yeah.

Leo: We buy our certs from them because their cert prices are so cheap for the EV certs.

Steve: Right.

Leo: But, I mean, that's a cert. That doesn't, you know, that's our security, not theirs.

Steve: Yeah. So anyway, what I want from my domain registrar is staid, stodgy, and stoic.

Leo: Yes.

Steve: I don't want a domain registrar that looks like Romper Room. And as I was putting that in the show notes, I thought, I wonder how many of our listeners will relate to Romper Room?

Leo: Oh, yeah.

Steve: You know, I think I'm beginning to date myself here a little bit.

Leo: I see Stevie. And I see Lorrie. I used to know Miss Nancy, our local Romper Room lady, actually.

Steve: So anyway, from a registrar I don't want entertainment and upselling. I just want something solid. Anyway, as we know, I chose Hover, and I've been very happy. And just to be clear, my choice was made years before Hover became a TWiT sponsor. So it wasn't like, you know, it wasn't after the fact. So in a filing Thursday, last Thursday, with the SEC, you know, our U.S. Securities and Exchange Commission, GoDaddy admitted that three serious security events, the first occurring three years ago in 2020, and the way they put it, you know, somehow lasting through 2022, were all carried out by the same intruder.

Now, okay. But they're also saying, but we don't know who. But we know it's the same. So I'm like, what? Anyway, they wrote: "Based on our investigation, we believe these incidents are part of a multi-year campaign by a sophisticated threat actor group that, among other things, installed malware on our systems and obtained pieces of code relating to some services within GoDaddy." And they said that their investigation was still ongoing.

The most recent event occurred last December, so just three months ago, when the threat actor gained access to the hosting servers GoDaddy's customers use to manage websites hosted by GoDaddy. They got into their cPanel hosting servers. The threat actor installed malware on the servers that "intermittently redirected random customer websites to malicious sites." Because, you know, that's what you want from your registrar. GoDaddy was unaware of the presence of this malware and learned of it from their customers, who were complaining that visitors to their sites were occasionally being redirected elsewhere.

So GoDaddy said: "We have evidence, and law enforcement has confirmed, that this incident was carried out by a sophisticated and organized group targeting hosting services like GoDaddy." They said: "According to information we have received, their apparent goal is to infect websites and servers with malware for phishing campaigns, malware distribution, and other malicious activities." Now, okay. Saying "hosting services like GoDaddy," you know, that sort of begs the question whether other hosting services

have been similarly affected. If so, you know, which ones? And by whom? Those questions remain unanswered.

It appears that the first of several intrusions took place in March of 2020, so fully three years ago, when a threat actor obtained login credentials that gave it access to employee accounts and the hosting accounts of roughly 28,000 of GoDaddy's customers. Fortunately, those hosting login credentials that were obtained for the 28,000 customers did not also provide access to the customers' main GoDaddy account. Otherwise damage would have been more severe. That first breach was disclosed two months later in May of 2020 in a notification letter sent to the affected 28,000 customers. The company said on Thursday it's responding, get this, responding to subpoenas related to that incident that the Federal Trade Commission issued in July 2020 and October 2021. So there doesn't seem to be any big hurry over in GoDaddy Land to do much of anything.

Then GoDaddy discovered another incident in November of 2021, two months after the threat actor obtained a password that gave access to source code for GoDaddy's Managed WordPress service. So beginning two months earlier in September of 2021, this unauthorized party used their access to obtain login credentials for WordPress admin accounts, FTP accounts, and email addresses for 2.1 million current and inactive - that is, previous - Managed WordPress customers at GoDaddy.

And these were not the first of GoDaddy's many problems. Through the years, security lapses and vulnerabilities have led to a series of suspicious events involving large numbers of sites hosted by GoDaddy. For example, back in 2019 a misconfigured domain name server at GoDaddy allowed hackers to hijack dozens of websites owned by Expedia, Yelp, Mozilla, and others and use them to publish a ransom note threatening to blow up buildings and schools.

The DNS vulnerability which was exploited by the hackers had come to light three years earlier, yet GoDaddy never took any action to mitigate the risk. Again, this is not the registrar you want. Also in 2019, a researcher uncovered a campaign that used hundreds of compromised GoDaddy customer accounts to create 15,000 websites that published spam promoting weight-loss products and other goods promising miraculous results.

Okay. So pushing back from this a bit, you know, the one question I had was how it was that GoDaddy could assert, through these more recent three attacks spanning the same number of years, that they had been repeatedly plagued by a single threat actor, yet somehow have no idea who this individual or group is. So I did a bit more digging, and I found that in their 10-K filing with the SEC they stated that the most recent December 2022 incident is connected to the two other security events they suffered in March 2020 and November 2021. Okay. Connected how?

This reminded me of what we recently saw from LastPass, where we were told that the second attack, the one remember where all of our backed-up LastPass vaults were stolen, was enabled by the initial intrusion. That was worrisome since it suggested to us that LastPass had not fully cleaned up after the first intrusion. In the GoDaddy case, they appear to be stating that they know that it's the same threat actor because information presumably obtained during the initial intrusion, three years ago back in 2020, was subsequently used in both 2021 and 2022.

Unfortunately this suggests, as with LastPass, that post-intrusion cleanup may have been minimized. And boy, given their track record and their apparent negligence, based on the actions that we've seen, who would be surprised by that? But in any event, the cleanup was ineffective. A full post-intrusion clean-up means that nothing that an intruder could possibly have obtained remains valuable once the clean-up is concluded. We know that didn't happen in the case of LastPass, and that also appears to have been the case for GoDaddy.

You know, as we've had occasion to note on this podcast, Leo, and you and I talked about it years ago, once malware has had access to a system, you can never fully trust it again. And I should really remove the qualifier "fully." You cannot trust any system after it's been compromised because you just don't know what could have been done. These days we have malware burrowing into our motherboard firmware to maintain persistence, even across wipes and completely reinstallations. So the only course of action then is to reflash the firmware, wipe the drives, rebuild from scratch, and change everyone's access credentials.

Yes, this is a huge nightmare in the case of a large sprawling enterprise, but there's really no choice. After GoDaddy's initial 2020 breach, either something lingered in a system that was never found, some latent advanced persistent threat presence, or they failed to rotate all of the keys and login credentials across the entire enterprise. Something remained, either malware tucked away in an unexamined corner, or someone's credentials that were never changed. Thus the same guys came back later for another dip, and a year later for yet another one. Wow.

Okay. Today and tomorrow the U.S. Supreme Court will be hearing initial oral arguments. And Leo, in your quick summary of the podcast you properly named the first of the two cases.

> **Leo:** Gonzalez v. Google. Yeah, I listened all morning. It went on and on and on and on and on and on and on.

Steve: Yeah, well, those attorneys do.

> **Leo:** Yeah.

Steve: Anyway, the U.S. Supreme Court's hearing oral arguments in a pair of cases which will open the door to allow the court to reexamine the now-famous and infamous Section 230 of the Communications Decency Act, which was passed into law by Congress 27 years ago, back in 1996. There are a crucial 26 words from Sector 230 of that law that are what enable our Internet's media companies to remain unresponsible, and some would say irresponsible, for the content that their users post online for consumption by others.

Those 26 words are: "No provider or user of an interactive computer service shall be treated as the publisher or speaker of any information provided by another information content provider." 26 words. And they mean, essentially, this blanket protection provides that none of today's media companies, the way this has been used to thwart any attempts at civil liability, is that none of today's media companies can be held responsible for the content that's being served by their technologies. Thus it serves as powerful and what has now become crucial protection for them. But many wonder whether it might have been taken too far.

The specific question that the cases address focuses upon the content promotion algorithms used by Google, for example, for YouTube, and also Facebook, Twitter, and others, to provide their users with more relevant content. So the question may be whether our social media companies have actually crossed the line to become publishers of this content the moment they involve themselves in that content's deliberate selection and promotion, even if that involvement is entirely algorithmic. The argument, then, is that they're no longer acting as passive repositories of user-provided content, and that the selections made by their algorithms are ultimately motivated by profit.

There's a cybersecurity law professor, Jeff Kosseff, he's with the U.S. Naval Academy, who wrote an entire book on Section 230, titled "The Twenty-Six Words That Created the Internet." And in some reporting by the Washington Post early last October - which is when the Supreme Court decided that they would hear the two cases which are now before them and for which they are now hearing these oral arguments today and tomorrow. Tomorrow is about Twitter. Today is about Google and YouTube. They quoted Professor Kosseff, saying: "The entire scope of Section 230 could be at stake, depending on what the Supreme Court wants to do." And, you know, although the stakes could not be much higher, the way these things go we won't have a decision anytime soon. Probably not till way later in the year, like toward the end of the year at the earliest. But this will certainly be one to watch.

And for their part, the plaintiff's attorneys say that applying the sweeping civil immunities created by Section 230 to algorithmic recommendations incentivizes the promotion of harmful content, and that Section 230 denies the victims of such content any opportunity to seek redress when they can show those recommendations caused injuries or even death. So this will be very interesting. And I've forgotten, Leo, where you come down on 230.

Leo: Oh, well, let me put it this way. You like the chatroom? You like the Discord? You like your forums. You like our forums. You like our Mastodon. If 230 is overthrown, all of those go away.

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

Leo: Yeah, all of them go away because right now I can't, and you can't, be sued for anything anybody posts on those forums. Even if it's defamatory or whatever, they're liable for it, not you. Which is reasonable; right? Furthermore, thanks to Section 230, if you take something down on your forums, and because it's racist hate speech, that person can't sue you either. And that's really important. It's the right both to publish and to moderate and not be liable. And because it's codified into law that way, you don't even have to go to court. You know, a justice, the judge would immediately say no, I'm sorry, he's protected by 230.

Steve: Right.

Leo: So if they strike it down or even weaken it in any way, you know, it's not Google and Facebook and Twitter who are going to suffer. They can defend themselves. They have lawyers by the fistful. It's you and me.

Steve: Well, so in this case we're glad that Supreme Court has a conservative bias at this point in time; right?

Leo: Well, they don't have a conservative bias. That's a misnomer. They're not originalist. They just make up whatever they want and then find something to justify it. I would be much happier if they were, yeah. But remember this is a 1996 law. Ron Wyden wrote it, and he was a very smart guy. And it was while they were passing the Communications Decency Act he said, you know, this could really screw up the Internet. We need to provide, you know, a safe harbor.

Steve: Protection, yeah.

Leo: Yeah. And so it's very, very important to the Internet. You quoted the exact right book. Jeff Kosseff's book is often referred to on This Week in Google. Jeff Jarvis is a big fan of it. I've read it. It's a very, very good book. And you read it and understand it. I listened to the arguments this morning. And you never can tell with the oral arguments in front of the Supreme Court because justices will sometimes play devil's advocate. Their actual opinions aren't always on display. But I was pretty encouraged by the questions they asked the counsel for the plaintiff. And I think they get how important it is. They even - one of the justices even said, you know, this could have a real impact on the economy. And then Justice Kagan, who I love and was very funny, said, you know, you don't have the smartest Internet brain sitting in front of you right here, so you'd better explain it to us. It was good.

Steve: So why did they even choose to take it up last October?

Leo: They could have let the Ninth Court decision stand because it upheld the Section 230 rights.

Steve: Right.

Leo: It was appealed, and you're right, that's the question is why did they take it up. And I think, you know, there probably is some reasonable discussion around this. What they're really battling over is not so much the right to publish or the right to moderate, but whether a recommendation algorithm is in some way now editorializing. And at first, I'll be honest with you, when I first read the facts of the case, I said, well, you know, that's actually a good point. You know, in a way Google's algorithm is choosing what to show. Isn't that Google creating content?

But I've since seen the light and been persuaded by a lot of smarter people than I, including Cathy Gellis from Techdirt who we're trying to get on the show tomorrow. She wrote an amicus brief for this. They also allowed multiple anonymous Reddit moderators to file an amicus brief, as did the EFF. Unfortunately, both the White House and the right, Josh Hawley and Ted Cruz, want this to be struck down, for different reasons, you know. But the wiser heads point out that it's all algorithmic. If you have a search engine, and you go to the search engine, what's on top of the search, unless it's completely chronological, is algorithmic.

Steve: It's the only reason that we all switched to Google, away from Alta Vista, when Google appeared.

Leo: Yeah, exactly. And the Reddit moderators say, no, we use algorithms to help us moderate. Algorithms aren't inherently bad. You might have an algorithm that's optimizing for profit, but as a result surfaces more controversial videos. But that's not the same thing as writing an article saying I think ISIS is fantastic. And so it's very risky, and I certainly hope the judges don't do this to slowly pare away at 230. It's only, as you say, it's only 26 words.

Steve: Right. And it is black and white at the moment.

Leo: It's very clear. It's I think one of the best written laws ever. It's kind of like a constitutional amendment. It's precise. It's broad enough to have lasted 20 years, 30 years. And but at the same time, you know, it's clear. And I think its intent is clear. And I'm hoping that the Court does not override what was clearly the intent of Congress when they wrote that law.

Steve: Yeah.

Leo: So let's, yeah, let's cross your fingers. I don't know if they're conservative, but let's hope they make the right choice.

Steve: So The Verge's headline was "It's Official: Twitter will now charge for SMS two-factor authentication. Only Twitter Blue subscribers will get the privilege of using the least secure form of two-factor authentication." And they were having fun with this. The Verge continued: "Now it's official: You can pay for the privilege of using Twitter's worst form of authentication. In fact, if you don't start paying for Twitter Blue ($8 a month on Android; $11 a month on iOS) or switch your account to use a far more reliable authenticator app or physical security key, Twitter will simply turn off your two-factor authentication after March 20th."

And the writer adds, he says: "I know which one I would choose. Good riddance to SMS is my feeling, given how common SIM swap hacks are these days." He says: "Heck, Twitter's own Jack Dorsey was successfully targeted by the technique four years ago. You don't want someone to get access to your accounts by proving they are you simply because they've stolen your phone number."

That's how Twitter is trying to justify this change, too, but I wouldn't be surprised if there's a simpler reason. It costs money to send SMS messages, and Twitter does not have a lot of money right now. The company had been phasing out SMS even before Elon Musk took over. Twitter's own transparency data shows that as of December 2021, only 2.6% of Twitter users had two-factor authentication turned on, and 74% of those users were using SMS as their two-factor authentication method.

Okay. So here's what Twitter posted and explained last Wednesday. Their blog was titled "An update on two-factor authentication using SMS on Twitter, by Twitter, Inc. We continue to be committed to keeping people safe and secure on Twitter, and a primary security tool we offer to keep your account secure is two-factor authentication. Instead of only entering a password to log in, 2FA requires you to also enter a code or use a security key. This additional step helps make sure that you, and only you, can access your account. To date, we have offered three methods of 2FA: text message, authentication app, and security key.

"While historically a popular form of 2FA, unfortunately we have seen phone number-based 2FA be used and abused by bad actors. So starting today, we will no longer allow accounts to enroll in the text message/SMS method of two-factor authentication unless they are Twitter Blue subscribers. The availability of text message 2FA for Twitter Blue may vary by country and carrier.

"Non-Twitter Blue subscribers that are already enrolled will have 30 days to disable this method and enroll in another. After March 20th we will no longer permit non-Twitter Blue subscribers to use text messages as a two-factor authentication method. At that time,

accounts with text message two-factor authentication still enabled will have it disabled. Disabling text message two-factor authentication does not automatically disassociate your phone number from your Twitter account. If you would like to do so, instructions to update your account phone number are available on our Help Center."

And finally: "We encourage non-Twitter Blue subscribers to consider using an authentication app or security key method instead. These methods require you to have physical possession of the authentication method and are a great way to ensure your account is secure."

Okay. So some other reporting I found stated that Twitter took this step because SMS two-factor authentication was being abused by fraudsters who would establish accounts using something called "Application to Person," or A2P, premium telephone numbers. Then when Twitter would send two-factor authentication texts to these numbers, the fraudsters would get paid. So it cost Twitter much more money than just a regular SMS to regular people. Estimated losses were claimed to be around $60 million a year from this.

Okay. So of course everyone's piling on Elon these days. And his decisions at Twitter have been a source of controversy. 74% of 2.6% is 1.95%. So as of the end of 2021, when we had those stats, 1.95% of all Twitter account holders were using SMS-based two-factor authentication. On the other hand, that's three out of every four of the Twitter users who use any form of two-factor authentication were using SMS, and the use of any form of two-factor authentication certainly prevents some amount of abuse. And even though SMS is not, we know, the best solution, it's still better than having none, and using it doesn't create any new vulnerability where none existed before. Unless I guess you were to, like, become dependent upon it and, like, had a crappy password because you figured, oh, well, two-factor authentication will protect me.

You know, so it's not something that can be relied upon nearly as much as one-time passcodes or security keys. So I don't think this is great news because it seems to me that it might end up causing Twitter users to simply disable all use of two-factor authentication without upgrading their existing SMS, you know, least of the three good authentication methods to one-time passcodes or a security key.

At around 450 million monthly users of Twitter, that 1.95% who have been using SMS-based two-factor authentication is 8.25 million SMS users per month. So that likely adds up, and I can see Elon wanting to cut cost. And if there's no way for Twitter to determine whether the phone numbers being registered are "pay to send" numbers, then I suppose he doesn't have much choice. On the other hand, a great many other large social media organizations offer SMS-based two-factor authentication, and they don't appear to have any similar problems.

In any event, I hope that those who need some form of authentication will move to passcodes at least, rather than just putting off, you know, all extra authentication when Twitter kills two-factor authentication a month from now. I think it's actually on March 20th, so a month from yesterday.

We have some good news. We knew it was coming. It has actually happened. And I've seen texts, or tweets rather, speaking of Twitter, from our listeners, wondering if they should move yet. "Maybe" is the answer. The Argon2 memory hard...

Leo: Woohoo!

Steve: ...PBKDF, yup, which promises to be far more resistant to brute forcing, is now available from Bitwarden and is present on "some" Bitwarden clients. And that's the key word. Before switching to it, since the switch must be made system-wide per user, you'll need to wait until, and make sure, that all of the platform clients, the Bitwarden platform clients you use, have been upgraded to support Argon2 and [crosstalk] record...

Leo: 2023.2, that's the version you need.

Steve: Exactly. That's the one you want, 2023.2.

Leo: I have it on my iPhone. I don't yet have it on Android. But you even have to have it on wherever you use it, on your desktops, on your plugins and all of that.

Steve: Yes, it's got to be in your browser extensions, and apparently it's not quite there yet.

Leo: You'll be blocked; right? You won't be able to use it if it's...

Steve: Correct. You will not be able to authenticate on that new device.

Leo: Right.

Steve: Six days ago a Bitwarden employee named Ryan, he posted to Reddit. He said: "For those curious as to why not everything is rolled out at once, each browser extension and mobile app needs to go through an approval process with their respective app stores. Please be patient. Usually the approval process takes about a week." So, you know, this is fresh news, but it's coming soon to Bitwarden platform clients near you.

Leo: That's the good news is that Bitwarden has approved the pull request, added it, and it is in the new version. Just wait till you get the new version. You will.

Steve: Right. And if you have it in iOS, then that's significant.

Leo: Yeah. I just got it a couple of days ago on iOS. I've been watching with great interest, as you might imagine. And I will switch as soon as I can do that safely, yeah.

Steve: So Mark Zuckerberg posted an announcement about a little change in Meta. He said: "Good morning and new product announcement. This week we're starting to roll out Meta Verified - a subscription service that lets you verify your account with a government ID, get a blue badge, get extra impersonation protection against accounts claiming to be you, and get direct access to customer support. This new feature is about increasing authenticity and security across our services. Meta Verified starts at $12 per month on the web or $15 per month on iOS."

Leo: Yow.

Steve: I know. That's exactly my feeling. He says: "We'll be rolling out in Australia and New Zealand this week and more countries soon." So, okay. Facebook is adding paid identity verification and more. So elsewhere in their announcement they wrote: "Some of the top requests we get from creators are for broader access to verification and account support, in addition to more features to increase visibility and reach. Since last year, we've been thinking about how to unlock access to these features through a paid offering.

"With Meta Verified you get a verified badge, confirming you're the real you, and that your account has been authenticated with a government ID." As also mentioned, I don't think they say it here, you have to be using your real name on your Facebook page, not some random handle. "Also you get more protection from impersonation with proactive account monitoring for impersonators who might target people with growing online audiences. Third, help when you need it with access to a real person for common account issues. Fourth, increased visibility and reach with prominence in some areas of the platform like search, comments, and recommendations. And, finally, exclusive features to express yourself in unique ways." And we don't know what those are.

So first of all, I reacted exactly as you did, Leo. 12 bucks a month on the web and 15 bucks a month on iOS strikes me as really expensive. It's not a one-time verification fee, which would seem reasonable. This is an ongoing cost, you know, $144 a year or $180 a year on iOS. And so I suppose it's not for everyone. If someone uses Facebook as a major platform, then I could see how it makes sense to pay something to obtain spoofing prevention and apparently higher visibility in search ranking results.

Leo: You don't get ad-free, though; right? I mean, it's not like - you only pay us seven bucks, and you get ad-free. I don't, you know, I don't really understand. And it's not for businesses. It's only for individuals. It's very strange.

Steve: Right, correct. That's not available for businesses.

Leo: Yeah, yeah.

Steve: They said "at this time."

Leo: Well, we'll see. We'll see. It's not going to generate 10 billion dollars a year, and that's what Mark's spending on VR right now.

Steve: No. No. Emsisoft, a company we've spoken of often...

Leo: I know, I remember that name.

Steve: Yeah. They basically provided us with a reminder of why simply having code signing is not and should not be sufficient to have antivirus and download protection warning silenced. And so the antivirus publisher Emsisoft has put out a public service announcement warning that threat actors are currently using fake Emsisoft code-signing

certs to sign their malware. This results in attacks appearing to come from Emsisoft's products, as well as to slip past anything that refuses to run unsigned software. So at some point I think what's going to happen, you know, code signing will become necessary, but not sufficient. At the moment, it's entirely optional, but mostly is there for user assurance. And I'm signing all of my apps now because it just seems like a good thing to do.

I know that when I'm - sometimes I'm digging around on the Internet, looking for some obscure thing because part of my life is still tied to DOS. If I see something on some download site, I will check to see if it's signed because, although as this little warning reminds us, it's not absolute assurance, but it's sure better than not having something signed. So, and it does, it certainly, if nothing else, it sends - it's a signal that AV and systems like Microsoft Defender can add to the conglomeration of other signals to decide what level of warning they want to provide the user.

Okay. DDoS attacks are always resource depletion or resource consumption of one kind or another. Today's modern DDoS attacks are typically no longer floods of TCP SYN packets like they were in days past. Those now seem quaint by comparison. Modern attacks are aimed less at consuming or clogging raw bandwidth than at asking web servers to generate more pages per second than they possibly can. Since modern websites are generally the front-facing surfaces of a complex content management system on the back end, which is driven by some form of SQL database, individual HTTPS queries have become much more computationally intensive than yesterday's serving of static web pages.

The previous contemporary-style DDoS attack-blocking record was set by Google Cloud, which last June reported blocking an attack rate of 46 million HTTPS requests per second. But that was then. Now, last week, Cloudflare has reported that it successfully fended off an attack that was 35% greater than that, mitigating a now new record-breaking and now setting HTTPS DDoS attack of 71 million requests per second. That's a lot of bots spread around the world all concentrating their fire onto a single target.

There are a growing number of strong website DDoS defenders. They include Akamai DDoS Mitigation, AWS Shield, Cloudflare's DDoS Protection, Google Cloud, F5's DDoS Hybrid Defender, Imperva DDoS Protection, and Microsoft's Azure DDoS Protection. Websites that pay to be located behind them are able to remain online even during an attack of such scale. That alone is somewhat astonishing. And an attack of this scale would utterly obliterate any other site that's simply "on the Internet."

The mitigation of attacks of such scale, while avoiding collateral damage to nearby resources, requires carriers of the attacking traffic which is bound for a site under an attack to block all traffic as far away upstream from the target as possible to prevent that traffic's aggregation as it moves from router to router approaching its destination. If we picture the Internet as a highly interconnected global network of individual routers, which is exactly what it is, each one forwarding traffic toward its destination, a useful overlay for this is the image of a great funnel, where incoming traffic is being funneled toward its target. In the model of a funnel, the closer we approach the funnel's neck, the greater the traffic burden becomes.

Since the physical implementation of this traffic movement are individual routers, the best defense against "too much traffic" is to cause attacking traffic packets to be dropped far out at the funnel's mouth. But doing this effectively inherently requires a large traffic provider. If the provider's network is not sufficiently large to allow the incoming traffic to be blocked before it has the opportunity to concentrate, then the provider's aggregation routers would be swamped themselves before it even gets to the user's web server. And many other of the provider's customers who are also being served behind those

aggregation routers, would have their access, their site access impacted by the collateral damage caused by a failure of the packet transport fabric.

An organization of Cloudflare's size, to name just one, has the advantage of operating at global scale. And when we're talking about handling attacks of this size, the network size is not only an advantage, it's a necessity. Since attacking bots are also globally spread, traffic bound for one customer's website will be entering the network of a global carrier such as Cloudflare at many peering points across the globe. So the moment an attack is detected, all of the provider's edge routing infrastructure can be informed of the attack and switched into an attack mitigation stance.

We talked many years ago about the sheer brilliance of the Internet's design, and with the original concept of autonomous packet routing being at the heart of this. That the original concept has withstood the tests of time, insane growth in usage and application, stands as a testament to those who created this system so long ago. But its great weakness is that it was never designed to withstand deliberate abuse. The idea that someone would flood the network with attack traffic was something that this system's gifted designers could never have anticipated. Even so, the Internet's basic architecture has been adaptable to incorporate such protections over time. So, wow. Hats off to them. And Leo, drinks up for me.

Leo: We do have DDoS - I actually shouldn't talk about our DDoS mitigation, should I. But we use it. And it's not Cloudflare. How about that? We might be using Cloudflare. We use somebody else. There are a number of people that do this, people with big fat pipes, basically.

Steve: Yup.

Leo: That's the key.

Steve: It's no mystery, though. Anyone can check to see where the traffic [crosstalk] sent to you.

Leo: Oh, they can tell? Oh, all right. Yeah, I guess you're right, come to think of it. So we use AWS. They have a very good DDoS protection solution, as well.

Steve: AWS Shield.

Leo: Yes. You mentioned it. And now I can tell the world. We use it. All right, Steve. On we go.

Steve: Speaking of DDoS attacks, I've often worried out loud here for at least the last couple of years about what would happen when malicious actors finally got around to focusing their evil intent upon, and commandeering for their nefarious needs, the truly countless number of Internet-connected, low-end IoT devices. Well, those worries are beginning to manifest.

Last year, from the summer, July, through December of 2022, Palo Alto Networks Unit 42 researchers observed a Mirai Botnet variant known as V3G4 predominantly leveraging

IoT vulnerabilities to spread. V3G4 targets 13 separate vulnerabilities in Linux-based servers and Linux-based IoT devices. The devices are commandeered for use in DDoS attacks. The malware spreads both by brute-forcing weak or default telnet and SSH credentials, and by exploiting known, but unpatched, firmware coding flaws to perform remote code execution on the targeted devices. Once a device is breached, the malware infects the device and recruits it into its botnet tribe.

And this is exactly what we've been worried about for years. Though it makes no rational sense at all, we know how difficult it is to even update big iron systems that need to be kept current, where there's a well-established notification and patching infrastructure in place to support that. Just look at the recent VMware ESXi fiasco. Those systems should have been readily updated. But as we know, they weren't.

So compare that to some random IP camera which was long ago installed and has since been forgotten. What about patching it? Good luck with that. We can't even keep our servers patched. Today, as I've often lamented, we have a literally uncountable number of gizmos and gadgets attached to the Internet. Why? Because we can. While most of those in our homes are safely tucked away behind the one-way valve of our NAT routers, and also hopefully on their own isolated network where possible, a great many, due to their role and application, have deliberately been given access to the public Internet.

In the present case of V3G4, Unit 42 tracked three distinct campaigns. Unit 42 believes all three attack waves originated from the same malicious actor because the hardcoded command-and-control domains contain the same string, the shell script downloads are similar, and the botnet clients used in all attacks feature identical functions. Yeah, that'd be enough to convince me.

Okay. So what does V3G4 attack? It exploits one of the 13 vulnerabilities. There's a CVE-2012-4869, which is a FreePBX Elastix remote command execution. There's a Gitorious remote command execution. There's a CVE-2014-9727, FRITZ!Box Webcam remote command execution. Mitel AWC remote command execution. There's a CVE-2017-5173, Geutebruck IP Camera remote command execution. Also a 2019-15107, Webmin command injection. Spree Commerce arbitrary command execution. FLIR Thermal Camera remote command execution. A 2020-8515 DrayTek Vigor remote command execution. Also same year, 15415 DrayTek Vigor remote command execution. In 2022, last year, 2022-36267 Airspan AirSpot remote command execution. Atlassian Confluence remote command execution. C-Data Web Management System command execution. 13 in total.

And notably, some of those CVEs were from 2012, 2014, 2017, and 2019. There's no reason to imagine that any of these problems will ever be repaired. And why would they be? The device is apparently working just fine. And who even knows whether the company that created it still even exists? A new trend we've observed is that companies are formed on the fly by pulling together the individual required resources. Devices are designed, they're manufactured, they're sold, then the entire briefly assembled organization dissolves, returning back to its original component parts. There is no one to call for updates. There is no follow-up. There is no accountability. There's no aftermarket, after-sale support. Yet an Internet-connected gadget can now harbor hostile code and be used, probably throughout the rest of its long service life, as one more tiny cog in a massive and untraceable global attack-launching platform. That's where we are today.

Again, in the case of V3G4, after compromising the target device, a Mirai-based payload is dropped onto the system and attempts to connect to the hardcoded command and control address. Once running, the bot terminates a large number of known processes from a hardcoded list, which includes other competing botnet malware families. Hey, I'm here now. You guys get out. You know? Now there's a new king of the hill.

A characteristic that differentiates V3G4 from most other Mirai variants is that it interlaces the use of four different malware XOR encryption keys rather than just one. This was clearly an attempt to make static analysis reverse engineering of the malware's code and decoding its functions more challenging. As I briefly noted earlier, when spreading to other devices, the botnet uses a telnet/SSH brute forcer that tries to connect using default or weak credentials and those 13 known vulnerabilities. Once set up and running, with a connection to the botnet's command-and-control, the compromised devices are then given DDoS commands directing their attacks. This variant offers TCP, UDP, SYN, and HTTP flooding methods.

The Unit 42 guys suspect that V3G4 sells DDoS services to clients who want to cause service disruption to specific websites or other online services, although the front-end DDoSing service associated with this botnet has not been identified at the time of Unit 42's report. So, you know, this is what was expected for a number of years was that eventually people were going to get around to getting serious about taking over our IoT devices and enlisting them in DDoS attacks. And we're now seeing a classic, perfect example of that happening.

So week after week I encounter news of malware stashes being found on this or that, or sometimes all, popular code registries and repositories. An example of such a piece of news from last week is that Checkpoint's research team detected 16 malicious JavaScript packages uploaded on the official npm registry. The researchers said that all packages were created by the same author, and were designed to download and run a hidden cryptominer on a developer's system. The packages pretended to be performance monitoring, so you'd expect them to use your computer's resources in order to determine how well a package is running. It, however, stays around afterwards, unbidden, to cryptomine in the background. All 16 of the packages have since been removed from the npm registry.

Anyway, so I just wanted to say that this is a constant flux. It's like that week after week, endlessly. I'm mentioning it this week because I don't mention all of this happening every single week in one form or another. Sometimes it's npm. Sometimes it's PyPI. Sometimes something else. Basically, wherever security firms are looking, they are now finding malicious packages. So I just wanted everyone to be aware that there is this constant flux of malware dribbling into the open source ecosystem. It's now another one of today's realities.

Leo: It's used everywhere, too, this package management system. We have Max. We have Homebrew. Every Linux distro has a package manager that downloads stuff. And security really is an afterthought. You know, I use a package management...

Steve: It's like, oh, hey, it's free. Grab this.

Leo: It's free. It's downloadable.

Steve: Grab this, you know, grab that. And the other thing is sometimes when you install something it comes with this massive list of dependencies; right?

Leo: Right.

Steve: Because you [crosstalk].

Leo: So those all download and install; right.

Steve: Exactly.

Leo: Right. You know, some of the package managers I use on Linux give you a chance to review the changes ahead of time. But even then, most of us just go yeah, yeah, yeah, whatever.

Steve: Leo, it's like a license agreement. It's like, oh, yeah, fine. What button do I push?

Leo: So it's page after page of code, of make file code and, you know, weird code. Ain't nobody got time to read that.

Steve: Nope.

Leo: So I'm not surprised. I think we've got to solve this, though. They're going to find a way to fix this somehow.

Steve: Yeah. And, you know, the problem is when you talk about closing it...

Leo: You can't.

Steve: Well, closing it is against the spirit of it being open.

Leo: Yeah, yeah, right.

Steve: Which is the whole point.

Leo: Right. I don't know how you do this, yeah.

Steve: So Patch Tuesday. That was last Tuesday. Many well-known publishers got in on the action. The industry was made aware of security updates released by Apple, Adobe, Git, Microsoft, and SAP. The Android project, OpenSSL, and VMware also released security updates last week. Microsoft patched 80, eight zero, vulnerabilities, including three zero-days; and Apple got a lot of attention releasing security updates that included a patch for an actively exploited Safari WebKit zero-day vulnerability. So everyone was told, you know, don't delay on that one. We know that the sometimes crucial mistakes many large and small organizations make is in ignoring these fixes. You know, if everyone kept their software patched we'd be seeing many fewer widespread problems, such as that VMware ESXi debacle which is still ongoing, by the way, more than 500 newly compromised systems just last week. So still happening, but slowing down.

As it turns out, however, and this is one reason that at least enterprises need to be a little careful, it wasn't all smooth sailing with this month's security updates. Microsoft has stated that some Windows Server 2022 virtual machines may no longer boot after installing the updates released last week. This issue, they said, only impacts VMs with Secure Boot enabled and running on VMware's vSphere ESXi 6.7 U2 and U3 or vSphere ESXi 7.0 point anything. The culprit is patch KB5022842 which, if installed on guest virtual machines running Windows Server 2022, may no longer start up. VMware and Microsoft are working to determine the cause. Interestingly, even though Microsoft says that only VMware ESXi VMs are affected, some admin reports point to other hypervisor platforms, including bare metal, also being impacted by this issue. So again, end-users should upgrade; enterprise users are always going to have to be on guard.

Last Friday, Samsung announced a new feature for, at the moment, only its Galaxy S23 series smartphones, called Message Guard. Now, the details are sketchy, and it sounds like it resembles Apple's "Blast Door" technology which Apple introduced back with iOS 14. Both technologies - Message Guard, which is Samsung's, and Blast Door, Apple's - are image rendering sandboxes.

We've often talked about the difficulty of safely and securely rendering images because image compression encodes images into a description that must later be read and interpreted in order to recover a close approximation of the original image. It's those image decompressing and rendering interpreters that have historically harbored subtle flaws that malicious parties have leveraged to create so-called "zero-click exploits," meaning that all the phone needs to do is display an image in order to have it taken over by a remotely located malicious party. So Samsung now has this technology added to its S23 series, and it has said that it plans to expand it to other Galaxy smartphones and tablets later this year that are running on One UI 5.1 or higher.

The addition of these technologies represents a maturation, I think, of our understanding of the problems we face. It is so easy to imagine, and every developer does, that any problem that's found will be the last one that will ever be found. And of course that's true, right up until the next problem is discovered. Experience shows that we're not running out of such problems anytime soon, if ever.

Okay. So it turns out that millions of Hyundai and KIA autos, which is to say approximately 3.8 million Hyundai and 4.5 million KIAs, are vulnerable to being stolen using just a bit of technology. And that indeed, once the method of doing so became common knowledge in some circles, Los Angeles reported an 85% increase in car thefts of those two brands. And not to be outdone in the car theft category, Chicago saw a nine-fold increase 900% in the theft of those cars.

Okay. So first, how was the news spread? Believe it or not, by something being called a "challenge" which has been heavily promoted on TikTok since last summer, July 2022. TikTok presented instructional videos showing how to remove the steering column cover to reveal a USB-A format connector which can then be used to hotwire the car.

Hyundai's and KIA's first low-tech response, which began last November, was to work with law enforcement agencies across the United States to provide tens of thousands of steering wheel locks. You know, a big red steering wheel locking bar has the advantage of letting TikTok-watching car thieves know that even if they're able to enter and start the car, aiming it will still present a problem. The fundamental problem surrounds a coding logic flaw that allows the "turn-key-to-start" system to bypass the engine immobilizer which is supposed to verify the authenticity of the code in the key's transponder to the car's ECU. In other words, no key is needed. This allows car thieves to activate the ignition cylinder using any USB cable to start and then drive off with the car.

Hyundai wrote: "In response to increasing thefts targeting its vehicles without push-button ignitions and immobilizing anti-theft devices in the United States, Hyundai is introducing a free anti-theft software upgrade" - oh, that's nice of them -

"to prevent the vehicles from starting during a method of theft popularized on TikTok and other social media channels."

Okay. So the software upgrade will be provided at no charge - you'd better believe it - for all impacted vehicles, with a rollout which began last Monday - a week ago yesterday - initially to more than a million 2017-2020 Elantra, 2015-2019 Sonata, and 2020 and 2021 Venue cars. All of the rest of the affected autos - and there were too many of them to list here - will be upgraded through the summer of this year. The upgrade will be installed by Hyundai's official dealers and service network throughout the U.S., and is expected to take probably less than an hour. Eligible car owners will be individually notified.

Hyundai's announcement explained that the upgrade modifies the "turn-key-to-start" logic to kill the ignition when the car owner locks the doors using the genuine key fob. After the upgrade, the ignition will only activate after the key fob is first used to unlock the vehicle, meaning that you can't break in first. That was the missing interlock which facilitated this hack in the first place. So the question remains, though, without a big red steering wheel locking bar, how would thieves without wheels know that your particular Hyundai or KIA is no longer vulnerable? Hyundai is solving this dilemma by supplying its customers, after they get the upgrade, with a convenient window sticker. And I would love to see what the sticker says, you know, like upgraded so the TikTok hack no longer works? Don't bother?

Leo: Can you put glue in the USB port? Would that help?

Steve: Well, and the problem is your car is going to get broken into before the bad guy is inside.

Leo: Yeah. Put a sign in the window that says "Glue in the USB port. Do not attempt."

Steve: Yeah. So Hyundai's providing a sticker, and I would love to see what the sticker says.

Leo: I'll show you this sticker we just got at Best Buy. You're going to like this. Remember to turn your computer off before 3:14:07 on 1/19/2038. I should send that to you.

Steve: That's brilliant.

Leo: As a Picture of the Week. I just saw this on Mastodon.

Steve: That's brilliant.

Leo: Hyundai's got a sticker that says what? Software upgraded. What?

Steve: I bet you won't be able to steal this car or something. I mean, like, what's it going to...

Leo: But would it - I bet it doesn't - is it really going to prevent that? I don't know.

Steve: Well, they're really going to put a sticker in the window. You know, and so but it only works for some. Unfortunately, there are some models that completely lack the engine immobilizer technology.

Leo: Ah, see? That's what I was thinking, yeah.

Steve: And so are unable, yes, they cannot receive the software fix which updates the missing immobilizer logic. So to address that problem, Hyundai will cover the cost of steering wheel locks for their owners. And, you know, this is the definition of a kludge. So far, all of this talk has been about Hyundai. But as noted, KIA has a similar problem.

Leo: Yes, same company, yeah.

Steve: KIA has promised to start the rollout of its software upgrade soon, but hasn't yet announced any specific dates or details. The U.S. Department of Transportation was the source of those stats about the number of affected vehicles, and also noted that these hacks have resulted in at least 14 confirmed car crashes and eight fatalities.

Leo: Oh, no.

Steve: So what do you want to bet that product liability and personal injury law firms are already rubbing their hands together over this quite significant screw-up? Wow. Okay.

Leo: Who says TikTok isn't useful? That's what I say.

Steve: So the astonishing success and the equally surprising performance of OpenAI's ChatGPT-3 Large Language Model AI means that a new phenomenon will soon be entering mainstream use. Leo, I'm going to take a sip of water. Why don't you tell our listeners...

Leo: Oh, good. I will. I'll tell you about Club TWiT while we get ready for - I'm dying to hear Steve's take on all this. This will be fascinating. We've been talking about nothing else on all the shows for the last couple of weeks. It's a hot topic.

Steve: And it's what gave our podcast the name today, "A Clever Regurgitator."

Leo: I figured that as much, yes. There have been lots of names for ChatGPT, including Mansplaining Machine, a Spicy Autocorrect. But I like the Regurgitator. That's good. That's good. And continue on talking about ChatGPT.

Steve: Okay. So as I started to say, the astonishing success and the equally surprising performance of OpenAI's ChatGPT-3 Large Language Model AI means that a new phenomenon will soon be entering mainstream use. I think that's absolutely clear. Right here on this podcast, thanks to Rob Woodruff's inspiration to enlist ChatGPT in assisting him with authoring that LastPass vault-deobfuscating PowerShell script, we've all witnessed first-hand just how significant these coming changes will be. And anyone who's been following the news of this may have continued to be somewhat astounded by what this technology appears to be capable of accomplishing.

I think that the most accurate and succinct way of describing what we're witnessing is that it is astonishing to see the degree to which a neural network using large language modeling, as exemplified by ChatGPT, is able to simulate intelligence. And I think that is the key concept to hold onto. ChatGPT is not itself in any way intelligent. It is a clever regurgitator of intelligence. One of the dangers, which we can feel present, is that this turns out to be a surprisingly subtle yet crucial distinction which is guaranteed to confuse many, if not most people who casually interact with this mindless bot.

After absorbing the historical global output of a truly intelligent species - namely, man - we have an automaton that's able to take our entire historical production, all at once as a whole, and quickly select from that massive corpus the right thing to say. It's able to choose it because that right thing has been expressed before, by man, in thousands of different contexts. So it appears intelligent because it's mimicking an intelligent species. A parrot in a cage who says "Polly wants a cracker" is more intelligent because it really does want a cracker. Although ChatGPT may be induced to express a desire, that's still nothing more than mimicry since it has previously absorbed all of humanity's past expressions of desire. It doesn't ever actually want anything because there's not actually any "it" there at all to do any wanting.

Again I come back to "Yes, what it does is astonishing." But that's only because it is the first thing we've ever encountered that's able to convincingly sound like us. But that's all it's doing. It's sounding like us. The parrot in its cage is extremely limited in its ability to sound like us. A sufficiently large language model neural network is potentially unlimited in its ability to sound like us. And if we can be certain of anything, it's that this simulation will be improving over time, especially now that this technology has left the lab and that capitalistic forces of commerce will be driving and funding further advancement. But nevertheless, in no way should "sounding like us" ever be confused with "being like us." A high-fidelity recording of Pavarotti may sound exactly like Pavarotti, but it isn't Pavarotti. It's just a recording.

Okay, so what got me started on this? It was an interesting experiment by some researchers at the company ANY.RUN who wanted to explore an aspect of ChatGPT's limitations. They wanted to see whether ChatGPT's otherwise impressive capabilities might extend to analyzing real-world malware. If so, it might make security researchers' lives more productive by allowing them to dump a load of code into ChatGPT and have it figure it out.

Their blog posting begins: "If ChatGPT is an excellent assistant in building malware, can it help analyze it, too? The team of ANY.RUN malware sandbox decided to put this to the test and see if AI can help us perform malware analysis. Lately, there's been a great deal of discussion about malicious actors using ChatGPT, the latest conversational AI, to create malware. Malware analysts, researchers, and IT specialists agree that writing code is one of ChatGPT's strongest sides, and it's especially good at mutating it. By leveraging

this capability, even wannabe hackers can build polymorphic malware simply by feeding text prompts to the bot, and it will spit back working malicious code.

"OpenAI released ChatGPT in November of 2022, and at the time of writing this article, the chatbot already has over 600 million monthly visits. It's scary to think how many people are now armed with the tools to develop advanced malware. So going into this our hopes were high; but unfortunately, the results weren't that great. We fed the chatbot malicious scripts of varying complexity and asked it to explain the purpose behind the code. We used simple prompts such as 'explain what this code does' or 'analyze this code.'" Okay. And then they go on with examples.

The short version of what they discovered is that ChatGPT did remarkably well when the researchers gave it toy code to examine. And it really did surprisingly well on that. But as the complexity of the testing code increased, there was a sort of "complexity cliff" they ended up going over, after which ChatGPT collapsed completely. And knowing what we know now, isn't that exactly what we would expect?

As a Large Language Model neural network, ChatGPT is not in any way even the tiniest bit sentient. Our limited-language parrot is more sentient. So ChatGPT is unable to "understand" anything at all. That means it's not going to be great at the true problem solving that reverse engineering complex malware code, or any code, requires. But reverse engineering code is very different from writing code. Thanks to the explosion of open source software, ChatGPT has previously ingested all of the source code on the Internet. That's a massive amount of real working code. And as we understand, it is able to select, regurgitate, and rearrange the code that it has previously encountered. But when it's asked to produce code that it hasn't previously seen, that's where things start to become fuzzy and where it starts making mistakes since, again, it's not really understanding anything about what it's doing. It's simply searching for a matching context amid all of the world's previously written code.

Last week I was corresponding with two of the sharpest minds I've ever had the privilege of knowing. And I was talking about the idea that I've previously shared here, which is that I think one of the things ChatGPT's surprising success at mimicry teaches us is that a good portion of the vaunted human intelligence we make such a big deal about having is mostly just repeating what we've previously encountered, and anticipating what's going to come next based upon what came next in the past.

Here's what I wrote to these two friends. I said: "If I look back over my creative life, there have been a few moments that I would say were truly inspired invention, where I created something from nothing, something that was actually new. But far and away 99.99999% of everything I do and have done has been wholly derivative. As it happens, I obtain immense satisfaction and even some joy from endlessly solving combinatorial puzzles. Thus I love electronics and coding."

Okay. So to wrap it up, I thought it was interesting and not at all surprising that, whereas ChatGPT can perform quite well at recombining what it's seen in the past to produce new and nearly functional code in the future, it is not going to be able to understand and explain the detailed operation of some piece of purpose-written malware that it has never encountered before. Though ChatGPT was initially a surprise, and though I'm sure that this technology is going to continue to improve over time, I believe that we now have a good foundation for understanding what it can and cannot do. And at least for the foreseeable future, it is at most a very clever regurgitator.

Leo: There's a good piece I'd recommend people to by Stephen Wolfram over on Wolfram|Alpha. It's called, I think, "How ChatGPT Works." And for the slightly mathematically inclined I think it would be very interesting. You know, he talks

about the initial kind of first approximation of how it works, which is basically autocorrect, using weighted values to predict the next word. It's a little more sophisticated than that. But it's essentially predicting the next chunk based on the statistical model. And it's quite interesting. Highly recommend it.

But yeah, I mean, it's not sentient at all. Obviously. And it's too bad because a lot of the press's focus was, especially with the Bing Chat, just based on the new ChatGPT-4 model, they were just needling it until it went crazy and then going, "You see? You see?" And it did feel like, you know, if it says "I love you," or "I hate you," or, you know, "I won't hurt you unless you hurt me," it sounds sentient. But it's honestly, it's really lost its marbles. And Microsoft's response to that was, well, after five questions we're going to reset. You can start over. You can't needle it into the point of insanity.

Steve: So Leo, I really do think we should not give it the nuclear launch codes.

Leo: No, probably not.

Steve: And I think resetting it after five questions sounds like a good idea.

Leo: Sensible, sensible.

Steve: And I hope that, you know, this is like maybe enough of a little bit of a freaky, yet still benign wakeup call that, you know, we're not in the future going to give anything the nuclear launch codes.

Leo: I think it helps us, after the initial wave of wow, understand a little bit more about what this is. It may pass the Turing test, but this is why the Turing test was a bad idea to begin with. That is not a measure of success really in a general artificial intelligence. We're still a long way off from that. Yeah, but don't give it the nuclear codes.

Steve: There are going to be a lot of people who are going to have long conversations into the middle of the night, you know, treating it like a therapist and a BFF.

Leo: There are, yeah. Like Eliza, yeah. It's like a good Eliza. Eliza was dopey, but this is surprisingly good, at least for the first hour or so. It really starts to get wacky after a while. Steve can go two hours and be coherent by the end. It's amazing. He's much better than ChatGPT. Steve's website, GRC.com, is the host of many fine things, including SpinRite, the world's best mass storage recovery and maintenance utility, currently 6.0. 6.1's on the way. You'll get it for free if you buy now. That's Steve's bread and butter.

He offers a lot of other free stuff there, including ShieldsUP! and so forth, Password Haystacks, lots of information. We talked the other day, somebody was talking about your DNS Benchmark program. And what's the InControl, the Windows 10 to Windows 11 stymie-er. We were talking about that on Sunday on Ask the Tech Guys. Lots of great stuff.