Not So Fast

Description: What danger is presented by the world's dependence upon GPS? And why is that of any concern? Has the sky fallen on all VPN systems? And why does the tech press appear to think so? Today's myriad network authentication options are confusing and incomplete. What does the future promise? Why might Apple have been erasing iCloud Keychain data? And what's actually going on between Google and the United Kingdom regarding the sunsetting of third-party cookies? What's the problem? Or is there one?

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-973.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-973-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He'll talk about GPS fuzzing, how it works, what one can do to avoid it. You've all heard about that VPN flaw that Ars Technica says makes all VPNs useless. Not so fast. Steve explains why it is not anything to panic about. And then, speaking of not so fast, Google has stopped progress on abandoning third-party cookies. Steve now knows why. He will explain all that and a whole lot more coming up next on Security Now!. Stay here.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 973, recorded Tuesday, May 7, 2024: Not So Fast.

It's time for Security Now!, the show where we cover the latest security and computer news and privacy news, and of course a little sci-fi and TV thrown in, with this guy right here, Steve Gibson, the arbiter of all that is good and kind. Hello, Steve.

Steve Gibson: Oh, well. I'll go for that.

Leo: Yeah.

Steve: Yeah. Hello, Leo. So here we are at the beginning of May. And as I promised, I did have some time to dig into the issue that came up actually two weeks ago when in the middle of the show you said, "Hey, Google just changed their plans on third-party cookies." And I said, "What?" Anyway, so we're going to talk about that. Today's episode is titled "Not So Fast," which as in that expression, "Not so fast, there." Which is what the UK is saying to Google.

But we're going to first look at what danger is presented by the world's current and growing dependence upon GPS, and why is that any concern? Has the sky fallen on all VPN systems, as the tech press has been reporting since yesterday, when a blog post...

Leo: Oh, good.

Steve: ...really went a little out of control.

Leo: I was really hoping, and I wanted you to explain Option 102 or whatever.

Steve: Option 121.

Leo: Oh, yes. I really want to know about that.

Steve: We will know all about that by the time we're done today.

Leo: Thank you.

Steve: Also a couple questions more from our listeners still bogged down in what is arguably a quagmire of network authentication options. So I'm going to spend a little more - that's continuing to come into crisper focus for me, so I figured let's - I'm going to spend a little more time on what's going on there. Also, we may have an answer to what Apple was doing with the iCloud Keychain deleting and what was going on, something that absolutely makes sense, so we're going to cover that.

And also, finally, as I said, I invested no little bit of time in - you'll hear the term "bureaucracy" used more times probably than any large word in this podcast because, boy, you know, I guess any kingdom that's been around as long as the United Kingdom and has continued to survive has also developed quite a system of bureaucrats, and they all want to weigh in on Google's plan. So anyway, I think another great podcast for our listeners. And a Picture of the Week that's kind of a hoot, too.

Leo: Oh, good. Always enjoy the Pictures of the Week. Well, Security Now! is ready to get underway. I hope you are, as well, boys and girls, cats and kittens, club members and others.

Steve: Well, the important work will be appearing shortly.

Leo: Hey, this is important work. Do not knock this work.

Steve: Now, we have a picture, a Picture of the Week, from somewhere, it looks like in the U.S. Southwest. There's no signs of any telephone poles or structures, so we're kind of out in the desert somewhere. And so one of the things that people want is they want their cell phones to work out in the middle of nowhere. And actually this is a problem I have with many movies these days, which seem to forget that it's necessary to have a

cell tower not too far away from where your cellular device is in order for it to get any connection. You know, we see people wandering out in the middle of literally nowhere, and they're on the phone. Unless the writers don't want them to be, in which case they're holding the phone up, you know, scanning around, trying to find a signal.

Well, the way we solve the problem of people wanting cell phone coverage wherever they are, yet nobody wanting to despoil the landscape as a means of providing it, is we come up with stealth cell phone towers. And I'm not sure how truly stealthful this is because it looks a little square to be a cactus. But I gave this picture the caption, "Oh, don't mind us. We're just putting the lid back on the cactus." Because this is clearly a cell phone tower cactus which is meant, I mean, it actually, you know, it's got the little extra, what do you call it, arm, off the side of the cactus, to make the whole thing look a little more cactus-like.

Leo: It's so funny.

Steve: And actually you can see some other cacti in the neighborhood that look decidedly less mechanical than this one.

Leo: All over Mexico you see these saguaro cactuses, and I guess the Southwest, as well.

Steve: Yeah.

Leo: So, you know, you see it with a hundred others, you probably wouldn't look twice. It's actually clever.

Steve: Yeah, it's certainly not an eyesore, looking like this thing would look like with the lid off, which we can see here because...

Leo: Right. The lid's off.

Steve: Yeah, the lid is off and the crane has lifted the lid off the cactus. Anyway, I just got a kick out of this. And I've seen fake palm trees, and I know that here on the so-called sort of now the famous 405 in Southern California there are power lines that run alongside the freeway, and every, like, very often there's a big cluster of cell equipment on the power lines because it's a perfect place for them to be, you know, there's already a right-of-way. There's some ability to run a service vehicle along the back and so forth. And many, many, many moons ago, back in the SpinRite - actually it was after SpinRite 2 because I remember I was working on SpinRite 3, I built a building in Aliso Viejo, you know, a corporate headquarters, 20,000 square feet, two stories, and 1.43 acres of land and so forth.

Leo: Holy moly. Wow.

Steve: Anyway, the cell companies came to me and said, hey, this building is like up on a point on a bluff looking out over this valley. You can make some extra money by letting

us put some cell things, like ringing along the edge of your roof. Well, you know what my answer was.

Leo: You said no?

Steve: The same answer - I said no.

Leo: Why?

Steve: This is a beautiful building. I'm not going to have warts of cell crap all over the...

Leo: I bet they're there now, Steve.

Steve: They are.

Leo: Oh, yeah.

Steve: I mention that because I drove by not long ago, looking wistfully up at the building, and there it was, just - I don't know. I don't think you could get more cell tower crap around the perimeter of this roof than there is there now. But not while I was in control. But immediately after I left, apparently. Anyway, such is the world, you know. And that's why I also have no ads on my site. Mark Thompson made a case, he said, at one point he said, "Steve, there's something wrong now with a website that doesn't have ads."

Leo: Yeah, what's wrong with you? Yeah.

Steve: No, thank you. Anyway, I wanted to start off this week by sharing an important piece of interesting news that's not Internet security-related, that is nevertheless potentially quite a big and serious issue in the real world. Last Thursday's headline in Wired was "The Dangerous Rise in GPS Attacks," with the subhead "Thousands of planes and ships are facing GPS jamming and spoofing. Experts are warning these attacks could potentially impact critical infrastructure, communication networks, and more."

Okay. So I thought that was interesting, got my attention. They said: "The disruption to GPS services started getting worse on Christmas Day." Meaning at the end of 2023. "Planes and ships moving around southern Sweden and Poland lost connectivity as their radio signals were interfered with. Since then, the region around the Baltic Sea - including neighboring Germany, Finland, Estonia, Latvia, and Lithuania - has faced persistent attacks against GPS systems.

"Tens of thousands of planes flying in the region have reported problems with their navigation systems in recent months amid widespread jamming attacks, which make GPS inoperable. As the attacks have grown" - no surprise to anyone - "Russia has increasingly been blamed, with open source researchers tracking the source to Russian regions such as Kaliningrad. In one instance, signals were disrupted for 47 hours continuously. On Monday, marking one of the most serious incidents yet, airline Finnair canceled its flights

to Tartu, Estonia for a month, after GPS interference forced two of its planes to abort landings at the airport and turn around." Talk about dependence on GPS. Apparently you just can't land anymore without it.

"The jamming in the Baltic region," they wrote, "which was first spotted in early 2022, is just the tip of the iceberg. In recent years there's been a rapid uptick in attacks against GPS signals and wider satellite navigation systems, known as GNSS (Generic Satellite Navigation), including those of Europe, China, and Russia. The attacks can jam signals, essentially forcing them offline, or spoof the signals, making aircraft and ships appear at false locations on maps." Which you can imagine might be even more damaging than just jamming outright. "Beyond the Baltics, war zone areas around Ukraine and the Middle East have also seen sharp rises in GPS disruptions, including signal blocking meant to disrupt airborne attacks." Which actually, as we'll see a little bit later, I think is the actual goal of this because of the degree to which drones are now using GPS.

Wired wrote: "Now governments, telecom, and airline safety experts are increasingly sounding the alarm about the disruptions and the potential for major disasters. Foreign ministers in Estonia, Latvia, and Lithuania have all blamed Russia for GPS issues in the Baltics this week and said the threat should be taken seriously. Jimmie Adamsson, the chief of public affairs for the Swedish Navy, told Wired: 'It cannot be ruled out that this jamming is a form of hybrid warfare with the aim of creating uncertainty and unrest. Of course there are concerns, mostly for civilian shipping and aviation, that an accident will occur, creating an environmental disaster. There's also a risk that ships and aircraft will suspend their traffic to this area and thereby affect global trade.'

"Joe Wagner, a spokesperson from Germany's Federal Office of Information Security, told Wired: 'A growing threat situation must be expected in connection with GPS jamming.' Wagner said there are technical ways to reduce its impact. Officials in Finland say they have also seen an increase in airline disruptions in and around the country. And a spokesperson for the International Telecommunication Union, a United Nations agency, told Wired that the number of jamming and spoofing incidents have 'increased significantly' over the past four years, and interfering with radio signals is prohibited under the ITU's rules." Gee. You think Russia is slowed down by a NATO agency, the International Telecommunications Union, saying, well, you shouldn't be doing that? Right.

"Attacks against GPS, and the wider GNSS category, come in two forms. First, GPS jamming overwhelms the radio signals that make up GPS and make the systems unusable. Second, spoofing attacks" - which actually are far more sophisticated - "can replace the original signal with a new location. Spoofed ships can, for example, appear on maps as if they're at inland airports." And actually that did happen recently. "Both types of interference have increased in frequency. Disruptions, at least at this stage, mostly impact planes flying at high altitudes and ships that can be in open water, not people's individual phones or other systems that rely on GPS.

"Within the Baltic region, 46,000 aircraft showed potential signs of jamming between August 2023 and March this year, according to reports and data from tracking service GPSJam. Benoit Figuet, an academic at the Zurich University of Applied Sciences who also runs a live GPS spoofing map" - there is such a thing - "says there have been an additional 44,000 spoofing incidents logged since the start of this year. Earlier this month more than 15,000 planes - earlier this month more than 15,000 planes had their locations spoofed to Beirut Airport, according to data that Figuet shared with Wired. More than 10,000 were spoofed to the Cairo Airport, while more than 2,000 had their locations showing in Yaroslavl, Russia, the data shows.

"Separate analysis from geospatial intelligence company Geollect shared with Wired showed that on April 16th around 55 ships broadcast their location as being over the

main runway at Simferopol International Airport in Crimea, Ukraine. The airport is around 19 miles inland from the Black Sea, where it's believed the ships were actually located." So, yeah, it's no longer possible to believe what GPS is showing you. You need to look out the window and see where you actually are.

"Zach Clements, a graduate research assistant at the University of Texas here in Austin, said: 'The biggest change in the past six months is definitely the amount of spoofing.'" As I said, spoofing is far more sophisticated and difficult than just jamming, and potentially far more dangerous. "He said: 'For the first time, we're seeing widespread disruptions in civil aviation, especially in the Eastern Mediterranean, the Baltics, and the Middle East. In prior years, there were reports of spoofing impacting marine vessels, but not aviation.'

"Clements says there appear to be three spoofers that can be traced back to Russia. One open source intelligence analyst, going by the pseudonym Markus Jonsson, has located jamming in the Baltics, and that which impacted the Finnish airline this week" - so that was the one that was causing them trouble - "to Kaliningrad and other Russian locations. One research group has suggested disruption near Poland impacted Russia's own GNSS system less than others." Not surprisingly, Russia doesn't want to hurt themselves. They just want to disturb everybody else. And "Russia has a long history of interfering with GPS signals, both within its borders and internationally. Russia's embassy" - not surprisingly - "in the UK did not respond to a request for comment.

"The disruptions can cause uncertainty and potential safety issues for airline pilots and their passengers." Yeah, no kidding. "A spokesperson for Eurocontrol, a European aviation organization with more than 40 countries as members, says its analysis shows disruptions are happening in the Eastern Mediterranean, areas around Ukraine and the Black Sea, as well as the Baltic states. During one week in March, 4,387 aircraft reported issues. The Eurocontrol spokesman says for the same time last week there were 2,646 flights reporting problems.

"The Eurocontrol spokesman says planes can fly safely without GNSS, but interference 'puts a higher workload on pilots and air traffic control.' A safety notice issued by the UK's Civil Aviation Authority this month says loss of GNSS, which is just, you know, general satellite-based navigation, can result in serious navigation issues, incorrect emergency terrain warnings that the plane is too low to the ground, and failure to various other systems."

And finally, in a NASA report detailing GPS incidents that was also published this month, one pilot said: "I have flown with crew members who were not fully aware of this problem." Other pilots said they had received "false terrain warnings" that caused them to pull up.

Leo: Ooh, that's not good.

Steve: Yes. And that pilots should have a "thorough review of jamming effects on the different aircraft systems" as part of their training. And here's the problem, of course. Because this is a relatively new phenomenon relative to when the pilots were trained, it may just be the fact that the pilots are trusting their avionics and not being sufficiently skeptical. So it does look like these GPS disruptions are coinciding with Russia's full-scale war in Ukraine.

And also it looks like Israel's attacks in Gaza have also been tied into this. As we know, disrupting GPS as part of electronic warfare has become common on Russia and Ukraine's battlefield as a way to try to limit the operation of drones. And while Iran launched a barrage of missiles and drones against Israel last month on the 13th, Israeli

GPS disruption designed to limit the impact of the attack also impacted mapping and taxi services, as well as food delivery. So here was an instance of Israel doing some GPS jamming which was somewhat indiscriminate, and the mapping and taxi services as well as food delivery within their own country took a hit as a consequence.

"Kevin Heneka," writes Wired, "the founder of cybersecurity company Hensec, whose work includes detecting GPS disruptions, says jamming and spoofing technology has become cheaper and smaller over the years, to the extent that individuals can install them in their cars to hide their own movements." That is, you know, you're blocking your own GPS receiver so your car doesn't know where it is. "However," Heneka says, "more sophisticated attacks use equipment that can cost huge sums." Yes. Anytime you're doing spoofing, as I said, spoofing is a whole 'nother level than just blanket jamming. He said: "In conflict zones, in military terms and in professional terms, this spoofing is very sophisticated, and it always goes hand in hand with jamming."

Okay. So since both the jamming and the location spoofing disruptions are enabled through the use of very powerful local radio transmitters, which overwhelm the reception of the authentic signals being beamed down from the GPS satellite systems in orbit, so long as you're not in the region of the Baltics, where Russia appears to have taken serious action to create major disruptions, the good news is, these attacks are inherently local in nature. You know, here in the U.S. we're not being affected by it at all, as is most of Europe. They are inherently very local.

But the problem for those who are in the region is that GPS and the wider GNSS, which again, Global Navigation Satellite System, have always been incredibly reliable sources. And not just of location, but also of time. They are master sources of essentially time of day. And as we know, when something is both very useful and has earned the reputation for also being very reliable, I mean, you know, these things are up in the sky beaming down at us, they end up creating a strong dependence. We end up becoming very dependent upon them. So many modern, non-military, commercial systems have become so reliant upon GPS that the deliberate disruption of that service for military purposes such as Russia has likely been perpetrating, can cause dramatic collateral damage.

The GPS system, which is put out by the U.S., was conceived quite a while ago, a little over 50 years ago, back in 1973. It took five years to package this in the first satellite that began launching at that time, and today we have 24 satellites up in the GPS constellation. They've been up and operating since 1993. And talk about depending upon something that's more fragile than we might want. Our phones and automobiles today only know where they are largely thanks to GPS signals coming from space.

Leo: I only know where I am thanks to GPS signals. Forget the car.

Steve: Yeah.

Leo: I can't drive without GPS.

Steve: And I'm sure, you know, a sports wristwatch, you know, health-tracking wristwatches are doing the same thing.

Leo: Oh, absolutely, very much.

Steve: And we have recently been talking about the militarization of space and the idea that having satellites attacking one another "up there" is no longer the territory of James Bond science fiction. You know, it's actually happening. In some cases robot satellites are there in order to repair others. But the same robot that can function to fix a broken antenna can also go over and break one off of some other satellite. So unfortunately they also have multiple purposes.

And unfortunately, as global political tensions increase, we can hope, and we need to hope, that no major powers having space-based military capabilities, nor the ability to kill satellites from the ground, believe that denying the entire world these benefits would create an advantage for them because it's difficult now to conceive of a world where GPS was just shut down, like destroyed deliberately by a power hostile to - it wouldn't even necessarily have to be hostile to the U.S. It could be because everyone's using GPS, killing it for everyone also succeeds in killing it for a specific targeted country.

Before GPS, the only way for something to know where it was, was through a system of inertial navigation. Inertial navigation, like its name suggests, is a closed system which relies upon the system's precise measurement of its own linear and angular accelerations. It integrates those over time to determine its velocities, and then integrates those over time to determine its velocities, and then integrates those over time to determine its position. Even though inertial navigation systems are still in use due to the nearly instantaneous position, and especially angular feedback that they provide, the errors that tend to creep in over time can only be eliminated with the use of slower but far more accurate input from the global GPS system.

I suspect Russia's primary concern is with the use of autonomous military drones, which may rely upon GPS to determine their in-flight location. But since the risks presented by GPS jamming, although they haven't been prevalent, and it hasn't been a big concern for airline pilots until recently, especially operating over there in the East in the Baltic areas, since jamming has been a possibility for some time, I suspect that the latest technologies are much more immune to GPS outages than those in Russia might wish.

Given all of the advantages and the advances made in vision and in real-time recognition, I would be surprised if the latest autonomous technologies were not able to fly nearly as well by sight as they can these days by GPS. They might well use GPS as a first choice, but use vision to detect location spoofing, while also being able to switch to pure vision if GPS should fail completely. And another likely strategy which, again, you don't worry about or deal with until it becomes a problem, is that since GPS signals will always be originating from above, would be to shield any GPS receiver and its antennas...

Leo: Oh, from below.

Steve: Yes.

Leo: Because the jammers are on the ground.

Steve: Exactly.

Leo: That's clever.

Steve: Yeah. So planes can do that because they're well above ground. Unfortunately, it's probably not practical for ships at sea.

Leo: Yeah, I mean, when you listen to ground air traffic control talking to an airplane, which I used to always do on United Channel 9, used to love to do that, they often have visual markers, you know, they say "Turn right at the Big Rock Candy Mountain" and things like that. I don't know if they still do that. I haven't listened in a while. But I bet they do. I mean, there's always - you always want redundancy in any system like that; right?

Steve: Yeah. And of course the problem is that, you know, we all - okay. I remember, Leo, when I guess this must have been in driver ed, we were supposed to go out and walk around our car to check all four tires.

Leo: Yeah, we don't do that anymore.

Steve: No.

Leo: Do you do that?

Steve: When was the last time anybody did that?

Leo: Pilots do that. And commercial jet pilots do that. And I think that goodness that they do. I think that's really great. But, no, I haven't done that to my car in a while. I figure if it's flat, I'll know. Right? It'll go frump, frump, frump.

Steve: That's right. That's right. But I do remember being told that's what we're supposed to do. So here we have a problem where GPS has been so reliable and relied on that I'm just hoping, I mean, in this NASA report last week where one of the guys said, you know, I've been with flight crews that just assumed that the GPS was telling the truth, even though they were suddenly being told to pull up because you're about to hit the Rock Candy Mountain, and that would not be good. But there's nothing there.

Leo: Pull up. Pull up.

Steve: So Leo, let's take another break, and then we're going to talk about whether the sky is falling on all VPN systems.

Leo: Yeah.

Steve: As the tech press seems to believe.

Leo: I was counting on you to cover this because I read the stories. Thank god you're covering it before I actually did the stories. Keep me out of trouble, please. Now, what's all this about VPNs, Steve?

Steve: Okay. So yesterday, Ars Technica got a little carried away in their reporting of what amounts to a clever hack that a Seattle, Washington-based pen testing firm known as the Leviathan Security Group posted in their blog. And of course the rest of the tech press picked up on it quickly, too. The blog posting carried the headline "How Attackers Can Decloak Routing-Based VPNs for a Total VPN Leak." And what I found curious was that they assigned - "they" meaning the Leviathan Security Group - assigned a CVE number to their discovery, even though nothing about this is a bug or a flaw.

Leo: Oh.

Steve: It's just a clever local exploit of a little-used feature of DHCP servers. Unfortunately, Ars Technica's headline for their story was headlined "Novel attack against virtually all VPN apps neuters their entire purpose." Agh, run away. Okay, which of course makes this sound more like the end of VPNs as we've known them. It isn't. Here's what's going on.

Okay. So going to do a bunch of propeller head cool stuff in order to get a real grip on this. Our PCs all interact with both internal and external networks through network interfaces. Most systems typically have a single physical network interface, or NIC; but it's possible for a machine to have more than one physical network interface with each interface connected to different physical networks. In that case, it's important for outgoing network traffic to know which physical interface any given packet should be routed out through.

To answer that question, our machines contain a routing table. The routing table performs a "most specific match" function, based upon the destination IP address. And in years past we've talked about Internet routing tables and all of this. So we've covered this in detail. But the key here is most specific match. And that all of our PCs, every one of them, pads, phones, you name it, anything that's networked using Internet protocol, IP protocol, has a routing table. Under Windows, for example, opening a command prompt and entering the command "route print" will display a list of the system's interfaces, followed by the IPv4 and IPv6 routing tables, respectively. And they're interesting, and you can get a sense for the fact that there's a lot going on under the covers that we don't appreciate, we normally don't even see.

Okay. So this set of network communication, that is, IP-based network communication, comes in so handy that in addition to true physical interfaces, many of our machines will have one or more virtual network interfaces. In fact, the so-called "localhost," you know 127.0.0.1, that's a virtual network interface that all stacks have. And, for example, the use of virtual machines has become very popular, and they create their own virtual network interfaces to talk to their host machine, as well as to the outside world.

Okay. So here's the main point: Many VPNs, like OpenVPN for example, operate by creating their own virtual interface in the hosting machine. It looks like and operates like any other network interface. But being a VPN (Virtual Private Network) which is used to transact privately with encryption, any packets sent out of that virtual interface are first encrypted, then rerouted out of an actual physical interface to be sent to the VPN's matching endpoint.

Since the typical VPN user, while using a VPN, wants all of their machine's traffic to be tunneled through the VPN, when the VPN tunnel is brought up, the VPN software dynamically edits the system's global routing table in such a way that, instead of the system's traffic by default being routed out through its normal actual physical interface, all of its traffic is instead routed to the VPN's software-created virtual network interface. This is the way that, deep down inside the guts of our machines, all of the traffic that's normally unencrypted suddenly becomes encrypted when we activate our VPN.

Essentially, it's like a man in the middle. It sticks a shim into our network so that all of the traffic that would normally just go straight out that physical interface instead is routed to the VPN. And that's done, as I said, by making just a slight change to the routing table so that all of the traffic, instead of going out the physical interface, goes to the VPN.

We need one other piece of information just to be certain that everyone's on the same page. DHCP stands for Dynamic Host Configuration Protocol. By default, when any networked machine boots up and gets itself going, it needs to be using an IP address for itself on its local network that's unique for that network. And it needs to know the IP address to which it should address packets bound for the outside world, in other words, the network's gateway IP. It may also want to know the IP addresses of some DNS servers that will honor its requests for domain name lookup.

It's the network's inward-facing DHCP server that answers all these needs. When any networked machine starts up, by default it will emit a broadcast packet onto the network announcing its presence and asking for any listening DHCP server to please provide it with all the information it requires to become a well-behaving citizen on the local network and to connect to the rest of the global Internet.

DHCP cleanly organizes the various types of information it can supply into, like, to the clients who are requesting it, by number. Each one of these is known as an "option," where the option number is a single byte, thus having a value from 0 to 255. Zero is a null option and can be used for padding; 255 is the marker for the end of the list of options. So the options are provided as a list of information terminated by Option 255, which of course, you know, is a byte of all ones.

So, for example, Option 1 provides the network's subnet mask to the requesting client. Option 2 specifies the offset of the client's subnet in seconds, that is, in real-time, from UTC, Coordinated Universal Time. Option 3 specifies a list of the IP addresses of routers on the client's subnet, what we know as the Gateway IP. Option 4 specifies a list of time servers which are available to the client. Number 6 provides a list of DNS servers for the client's use. And, you know, there's a bunch of them, all kinds of different things that have been added through the years. And there are even some surprises. For example, options 69 and 70 provide the IP addresses of SMTP and POP3 email servers, which I thought was kind of cool. We're all used to specifying those ourselves; but back in 1997, when this was first created, that information was available via DHCP.

Something else that DHCP was able to provide is the source of today's trouble. The RFC's definition for Option 33 defines it as the "Static Route Option" and says: "This option specifies a list of static routes that the client should install in its routing cache." Okay, now, everybody who's been paying attention and enjoys networking stuff just went "aha" and knows what the problem is.

This thing continues: "If multiple routes to the same destination are specified, they are listed in descending order of priority. The routes consist of a list of IP address pairs. The first address is the destination address, and the second address is the router for the destination." Again, if some of you just said "Oh, crap!" that would be the correct reaction. What this means...

Leo: And it would mean they're paying attention. Good job.

Steve: That's right. That's right. What this means is that the response from a DHCP server can be used to mess with a machine's routing table. And as we noted earlier, a machine's traffic is routed to the VPN's virtual interface through a dynamic modification of the machine's routing table.

Now, as it happens, Option 33 is not really the problem because it was defined back in 1997 when IP networks were all class A, B, or C. That meant that networks were defined to always have exactly one, two, or three bytes of host machine addresses. As we know, this was extremely wasteful of IP addresses for networks falling into intermediate sizes. So something known as CIDR, C-I-D-R, which stood for Classless Inter Domain Routing, was adopted. That's what we have today, where the network mask can have any number of contiguous bits set, thus allowing scaling of networks by factors of two, all the way from one machine, well, technically up to 4.3 billion, but no one network has that except the Internet itself.

Okay. So the adoption of CIDR obsoleted Option 33, forcing its replacement five years later in 2002 under the guidance of RFC 3442 which introduced Option 121, which allows for exactly the same thing, but under the specification of classless static routes.

Now, I mentioned that I was surprised that these Leviathan Security Group guys had arranged to get a CVE assigned for this, since technically this is a feature, not a bug. And all the way back in 1997 the fundamental vulnerability of DHCP was quite well understood. Again, 1997, Section 7 of the original RFC 2131 dated March of 1997, is titled, it was Section No. 7, "Security Considerations." It says: "DHCP is built directly on UDP and IP, which are as yet inherently insecure. Furthermore, DHCP is generally intended to make maintenance of remote and/or diskless hosts easier. While perhaps not impossible, configuring such hosts with passwords or keys may be difficult and inconvenient. Therefore, DHCP in its current form" - which, by the way, is the form it has today in 2024 because, you know, if it's not broke - "in its current form is quite insecure," says the RFC from 1997.

Leo: Wow.

Steve: They said: "Unauthorized DHCP servers may be easily set up. Such servers can then send false and potentially disruptive information to clients such as incorrect or duplicate IP addresses, incorrect routing information including spoofing routers, et cetera, incorrect domain nameserver addresses to spoof nameservers, and so on. Clearly," they wrote, "once this seed information is in place, an attacker can further compromise affected systems."

Okay. So here's how the Leviathan folks describe the attack they've devised by abusing Option 121. They said: "Our technique is to run a DHCP server on the same network as a targeted VPN user and to also set our DHCP configuration to use itself as a gateway. When the traffic hits our gateway, we use traffic forwarding rules on the DHCP server to pass traffic through to a legitimate gateway while we snoop on it. We use DHCP Option 121 to set a route on the VPN user's routing table. The route we set is arbitrary, and we can also set multiple routes if needed. By pushing routes that are more specific than a /0 CIDR range that most VPNs use, we can make routing rules that have a higher priority than the routes for the virtual interface the VPN creates."

As we know, because that means it's a more specific route, so the routing system will always route a more - will always take the most specific route available. So by doing something, creating a network smaller than the /0, which is the everything, the routing table ends up routing to the intercepting DHCP server rather than to the user's VPN. They said: "We can set multiple /1 routes to recreate the 0.0.0.0/0 all traffic rule set by most VPNs.

"Pushing a route," they wrote, "also means that the network traffic will be sent over the same interface as the DHCP server instead of the virtual network interface. This is intended functionality that is not clearly stated in the RFC. Therefore, for the routes we push, it is never encrypted by the VPN's virtual interface, but instead transmitted by the network interface that is talking to the DHCP server. As an attacker, we can select which IP addresses go over the tunnel, and which addresses go over the network interface talking to our DHCP server." So in other words, they're able to literally select by destination IP. If they don't want everything they can say, oh, just give us this chunk of your traffic. You think it's going through your VPN, but it's not.

They said: "We now have traffic being transmitted outside the VPN's encrypted tunnel. This technique can also be used against an already established VPN connection once the VPN user's host needs to renew a lease from our DHCP server. We can artificially create that scenario by setting a short lease time in the DHCP lease, so the user updates their routing table more frequently. In addition, the VPN control channel is still intact because it already uses the physical interface for its communication." That is, you know, the control channel meaning the channel to the remote end that is outside of the tunnel. They said: "In our testing, the VPN always continued to report as connected, and the kill switch was never engaged to drop our VPN connection." Meaning there was never a panic that the VPN was concerned that it was being intercepted and so shut things down."

So then, to their credit, they raise the question that we've had all along, by asking "Is TunnelVision a vulnerability?" And I appreciated their answer. They wrote: "This is debatable. We're calling it a technique because TunnelVision doesn't rely on violating any security properties of the underlying technologies. From our perspective, TunnelVision is how DHCP, routing tables, and VPNs are intended to work.

"However, it contradicts VPN providers' assurances that are commonly referenced in marketing materials. In our opinion, TunnelVision becomes a vulnerability when a VPN provider makes assurances that their product secures a customer from an attacker on an untrusted network. There's a big difference between protecting your data in transit and protecting against all LAN attacks. VPNs were not designed to mitigate LAN attacks on the physical network, and to promise otherwise is dangerous.

"In our technique, we have not broken the VPN's cryptographically secured protocol, and the VPN is still fully functional. An attacker is instead forcing a target user to not use their VPN tunnel. Regardless of whether we classify this as a technique, VPN users are affected when they rely on assurances that a VPN can secure them from attackers on their local network." And finally...

Leo: Hmm. Interesting. That is one of the primary uses, isn't it, for a coffee shop and their open WiFi networks.

Steve: Exactly.

Leo: Yeah, okay. But this has been around forever, so.

Steve: Yes, exactly. And they finished: "As for what systems are affected, the short version is everything except Android."

Leo: Isn't that funny.

Steve: Uh-huh. Android doesn't support Option 121. So it's completely excluded from these attacks.

Leo: Isn't that wild.

Steve: They wrote: "In our testing, we observed that any operating system that implements a DHCP client according to its RFC specification and has support for DHCP Option 121 routes is affected. This includes Windows, Linux, iOS, and MacOS. Notably," they wrote, "it does not affect Android as they do not have support for DHCP Option 121." Which, you know, really is interesting.

Leo: I wonder why not, yeah.

Steve: I do, too, because I did some digging, and there have actually been instances where Android's lack of Option 121 support has caused problems for Android users because it turns out this is not obscure, Leo. This is the first time we've ever talked about it on the podcast because it's just never come up. You know, we've covered DHCP in depth in the past.

Okay. So just to be clear about the scope of the danger presented by the potential abuse of DHCP's Option 121, this is strictly a local LAN-side attack. But Leo, as you correctly point out, you know, we do operate in essentially LAN networks where we're assuming a VPN is going to trust us where untrusted peers are on the same LAN we are. So that's a thing. The attacker needs some means of defeating the network's actual DHCP server. Since DHCP clients will and do accept the first reply to their query, simply being faster to reply is typically all that's needed.

And, you know, as we know, most routers use the slowest chip that the manufacturer was able to get away with. Boy, I tell you, those web interfaces on routers, it's like, okay, I click the button, hello, hello. Should I click it again or just wait? You know? So the point is it's not going to be quick to fire off a DHCP reply because it doesn't need to; right? That's going to be way down the priority queue of traffic that it needs to deal with. So an attacker probably doesn't have much difficulty being able to respond with DHCP queries faster. So it's definitely conceivable. Also, in an enterprise environment, that if you had somebody untrusted on an enterprise network, that would be a problem. And it also turns out that Option 121 is not the least bit obscure in the enterprise. Turns out it's under heavy use.

I found two little samples through a quick search. A posting over on Stack Exchange says: "I'm running OpenVPN on a CentOS 7 server. The DHCP server on the LAN uses Option 121 to tell other devices to use this CentOS server if they want to get to the VPN subnets the OpenVPN server is connected to. This works great. The problem is that this CentOS server is getting these same routes from the DHCP server, which breaks things." And then he goes on to talk about how he can manually remove the static routes that the CentOS server is receiving from DHCP. But my point is here's an example of where

Option 121 is being used to inform machines on the LAN where to route the traffic they want to go through the CentOS 7 server's VPN subnets. So it's very useful for that.

And also, just as recently as last Tuesday, someone posted to the what I have to categorize as the embarrassingly useless Microsoft answers forum. I don't know if anybody has ever seen any of the crap that is there. But, you know, if Microsoft really wants to lead in AI, they should remove whatever poor humans they have that are being forced to respond to forum postings there and put ChatGPT 12 or something in there instead. It is, I mean, it is excruciatingly bad.

Anyway, someone posted, and needless to say they got no useful answer: "When connected to my office network, its DHCP server," meaning his office's network DHCP server, "will use Option 121 to assign three different networks to be reached using a router which is not the default gateway. This works absolutely. The networks appear in my routing table in active routes. Everything works. Networks are reachable." Anyway, so he wrote that, and I just grabbed that as a little snippet of another example of like Option 121 is really out there, and it turns out has really been useful, as I said.

He goes on to explain at some length. He's complaining that when he boots his PC without any network connectivity, then it has a problem. Uh, yeah. That would be a problem. So anyway, I wanted to point this out, again, that this DHCP option is in heavy use within more complex corporate networks. What that means is that simply, like, blacklisting Option 121 is not viable. In my opinion, it would be extremely unlikely for anyone at home to ever have anything to worry about, though it's still instructive to paint a picture.

The way I can see this might occur to somebody at home would be if some malicious device were connected to a residential network and wished to capture all of the user's traffic, whether tunneled through a VPN or not. By being the first device to respond to any DHCP query, such a malicious device could establish itself as the network's gateway to receive, inspect, and forward all traffic from the network's many machines. And then, by additionally using Option 121, such a device could use that to also insert entries into the user's routing table to prevent their VPN, if any, from tunneling the user's traffic. Even though the VPN would show that everything was working and the user's traffic was protected, none of it would be. The VPN tunnel would be up and established, but it would not be carrying any of the user's traffic.

Since there are many environments where Option 121 is not needed and is never used, like probably most of ours at home, I think it would be nice for our operating systems to provide the option, like, to hard disable it. But I dug around, and I couldn't find any indication that that's being done.

Leo: Yeah.

Steve: I would imagine the Windows firewall could be configured to just, you know, to look for any incoming DHCP port, what is it, it's been so long, is it 163 is DHCP?

Leo: No idea, yeah.

Steve: I don't remember now the port numbers.

Leo: So the best mitigation would be to turn off Option 121, but that's not an option.

Steve: No.

Leo: What happens now? Unless can VPN software be updated to have that as a feature?

Steve: The problem is this gets in underneath the VPN software.

Leo: Yeah, yeah.

Steve: The VPN software, I mean, I suppose it could be updated to monitor the routing table and proactively determine whether or not it's been rerouted. So that's certainly something that could be done. Right now the VPN, when you bring up the VPN tunnel, it inserts a new default route for everything, and points it at its virtual interface so that it receives everything. What it would need to do would be to - and I guess it actually could - would be to send itself a test ping.

Leo: Ah, there you go.

Steve: From an IP in the user's IP space and verify that its virtual interface receives that ping.

Leo: Right.

Steve: If it doesn't receive the ping, that tells it something has interfered with the routing between the user's localhost IP and its own interface. So, yeah, that would be a cool feature for a VPN to add.

Leo: And meanwhile there's not really a mitigation, is there.

Steve: No. No. And I think your use case is exactly the right one, Leo, because, you know, where do people deliberately bring up a VPN? It's when they're in a hotel, in a caf, in any untrusted environment. And they don't want to be sharing their traffic with everybody else.

Leo: Yeah, yeah. I wonder if commonly used hacking tools like WiFi Pineapples and stuff are able to do this. They probably are. I mean, it's been around for 30 years.

Steve: Yeah, but, well, oh, so you mean whether they're able to perform the hack?

Leo: Yeah.

Steve: I bet that, you know, that intercepting...

Leo: That seems like something you'd build in.

Steve: Well, and intercepting DHCP is such a juicy target.

Leo: Yeah.

Steve: I mean, I'll bet you that hacking tools actively have DHCP server spoofing and are able to get a response out immediately.

Leo: Interesting. Wow. This is good stuff, thank you. Because this has been everywhere, this story.

Steve: Yeah.

Leo: And I was really curious what you thought of it.

Steve: So it's a problem. It's not, again, what are you going to do with a CVE?

Leo: Right.

Steve: Like, hello. Like, okay. I mean, maybe that gets it more attention.

Leo: It gets the word out; right?

Steve: Yeah. Unfortunately, apparently GPT something is able to read the CVE and immediately design a hack that the script kiddies can then use. So, great.

Leo: Would you like to take a break? Is that what you're looking at me like that for? I know that look. We will have more with Mr. Gibson in just a little bit. You know, every week there's a story or two that in my mind, and I bet your mind, too, you go, I wonder what Steve has to say about that. That's why we love you, Steve. And that's why we listen to the show. We're so glad to carry the show. On we go with Mr. G.

Steve: Okay. So a bit of feedback. Dave Brenton tweeted: "Mr. Gibson. Quickly may I say, as a machine language coder, I admire your work in that area. I'm a SpinRite owner/user and longtime fan since near the beginning of Security Now!. My question is about security keys. I hope this is not too long a question." And it wasn't.

He says: "I'm about to make the transition to YubiKey, and so I intend to purchase two, to have a safe fallback in case of loss. I'm also planning to convert the wife over to the Passkey world. My question is can the Passkeys be paired across two user accounts, thereby ensuring recovery in case of loss with only three keys? My mental model says it

made sense, but I do not know for sure. One, can the same key be applied to two different people? Two, to assure full backup protection, can all three keys be coded into both users? It may be a silly notion, but could it work? Or should I just buy four keys to begin with? Thank you for all your good work and propeller-head installments. On to 999 and beyond."

Leo: Yes.

Steve: Dave. Yes. And I said at the beginning of the show on Episode 973 we are closing in on 999.

Leo: We are.

Steve: Yet we're no longer fearful of that fatal number.

Leo: Made me sad. Hey, before you get to the answer, I just want to, well, actually do the answer. And then I want to ask you about machine language and assembly. I had some questions.

Steve: Well, cool.

Leo: Yeah. Go ahead.

Steve: Okay. So I chose to share Dave's question because it so perfectly demonstrates the near total mess the user authentication world has fallen into today.

Leo: It really has. It really has.

Steve: It is just a catastrophe. I'm hopeful this may just be a transition phase. But truth be told, all of our collective experience also leaves me feeling somewhat skeptical. I worry that all we have done by having the FIDO Group lower the bar for entry from requiring physical key dongles to allowing pretty much anything else - smartphones and PCs running simple software Passkey clients - is to expand upon the number of available options, with an additional and, difficult as it is to believe in this day and age, not very well-thought-out system. And we've added this new and not well-thought-out system without removing any of the previous options.

Have traditional username and passwords been replaced? No. Are they ever going to be? Not in this lifetime. Have the "I forgot my password" links gone away? No. Are they ever going to? No. What about those time-based one-time passcodes? Are they going away? No. Any plan for that? No. What about OAuth, which brings us the "Log on with your Google or Facebook or some other account?" Have those been obsoleted and removed? Nope. Can they be? Well, not easily, since many sites only know their users thanks to their redirection through another web service's authentication.

And so to this pile of existing half-baked remote network authentication solutions we are now adding Passkeys, a mysterious new solution that its designers all say is amazing and

far more secure, which works sort of like magic right up until it doesn't work at all. And when that happens, what do we do? Well, we fall back to "Send me an email."

What we've wound up with is the well-known and often observed phenomenon of "solution spread." We invent a better idea than what we had before. Perhaps it's because the times have changed and the older solutions are no longer adequate. Or perhaps we have more technology and available processing power than we had before, so new solutions are available than were previously. But the problem is, we rarely are able to kill off the things that came before. Why? Because by the time we can do something more, too many people have come to depend upon the previous solution, and the one before that, and the one before it.

And this solution-spread doesn't just apply to the authentication domain. Just look at Windows. Without getting bogged down into the details, every few years Microsoft comes up with a new and much improved way of writing applications for their Windows OS. And they promote the hell out of it, explaining how and why it's so much better than everything that came before. And do they then kill off the previous ways of programming Windows? No. Of course not. They can't. They were once promoting the hell out of those previous solutions, and they got lots of people onboard using them then. So even though they no longer love them and are urging everyone to use the new system, that never happens. I've heard Paul over on Windows Weekly saying that the original Windows API, Win32, should have died off long ago.

Leo: Oh, yeah.

Steve: That's what all of my Windows are written in; you know? And not just mine; a gazillion others, as well. And that's "gazillion" with a "G." I am certain Paul knows that Microsoft will never abandon Win32. They can't, any more than websites will ever be able to stop offering username and passwords with an "I forgot how" email link.

So just to be clear, the industry has added a bright and shiny additional way for people to log into their accounts. But none of the existing ways are, or will be, removed. Remember that today in 2024, only one out of every three Internet users is using any form of password manager. I really don't know what the rest are doing. Perhaps these are the people whose iOS and Android support for Passkeys is mostly aimed at. You know, these people don't know, don't understand, and don't care about their online identity. So when Apple or Google comes along and asks, "How would you like to log in instantly with Passkeys and never worry about another password?" well, that sounds great.

But that's not Dave, our listener whose questions launched me into first taking a bit of a rant into a wider view of where we stand today. So let's look at Dave's situation. Dave says he's planning to convert his wife over to Passkeys. I'm sure he means that he would like to have his wife begin to use Passkeys, since it's not possible to "convert over" to Passkeys in any meaningful way when so few websites offer the option at all. The caution there, since we do not yet have Passkey transportability, is to be careful about which app is holding a site's Passkeys. As I mentioned last week, iOS, Windows, Android, and now an increasing number of traditional password managers will all be vying to be the app that generates the Passkey to be provided to a website. Since only that app will then be able to authenticate the user to that site with a Passkey, the only sound strategy will be to only and always use a single platform for Passkeys.

This issue, and Dave's other questions, require a quick bit of foundation about the operation of Passkeys. When an application prompts its user about whether the user wishes to have it create a Passkey, that's exactly what's happening. The application

generates a cryptographically strong secret and private key, which never leaves the application and which the application guards carefully. From that closely held private key it then generates a public key, and only the public key is sent to and retained by the website. In the future, that website will use the public key it holds to verify the signature of a challenge that it sends to the user's Passkey authenticator.

So my point here is that, today, there is no provision for these private keys, which were generated internally and have ever since been guarded by the application, to ever leave that application's control. And a security-conscious organization like Apple can make the defensible claim that since all of the Passkeys' security derives from the "secretness" of these private keys, which is crucial, no other application, including its user, can or should be entrusted with their stewardship, with the stewardship of the Passkey's private key. Since this represents a powerful platform lock-in, it's not at all clear to me that Apple will ever allow for Passkeys export.

That being the case, I think that a very strong case can be made for only ever storing Passkeys in a third-party Passkeys client, such as a browser extension. In theory, it ought to be possible for a website to allow its user to replace one Passkey with another. So if Apple or Android were to inadvertently become the generator and holder of a Passkey, if a website supported Passkey replacement, it should be possible to migrate away from one Passkey application to another. And I was thinking about this. If a website doesn't explicitly allow you to migrate between Passkeys, hopefully it allows you to delete a Passkey, in which case your account would not be associated with one, and then you could reassociate it with a Passkey from the provider that you're wanting to switch over to. So the real point here is that it is the application that generates the Passkey. It is never something that we're able to supply from the outside.

So just to put a bit of frosting on this discussion before we talk about the platforms with hardware authentication dongles, I wanted to share a few points from Google's Chrome FAQ. This is Google's Chrome browser FAQ about Passkeys. They start off, of course, with all of the glowing bits. Under "Manage Passkeys in Chrome," they say: "You can use a passkey to sign in easily and securely with just a fingerprint, face scan, or screen lock. Passkeys are a simple and secure way to sign in to both your Google account and all the sites and apps you care about without a password. You may be asked to sign into a website with a Passkey or create one to improve your account's security." And then they have a little tip: "Passkeys are built on industry standards, so you can use them across many platforms."

Leo: Gotta love those industry standards.

Steve: Oh, Leo. That's the happy news. That all sounds terrific. And of course we ask here, what could possibly go wrong? Well, here's what Google has to say about that. Under "Store Passkeys in Windows" they said: "If you have Windows 10 or up, you can use Passkeys. To store Passkeys, you must set up Windows Hello. Windows Hello does not currently support synchronization or backup, so Passkeys are only saved to your computer. If your computer is lost, or the operating system is reinstalled, you cannot recover your passkeys." Whoops.

Or "Store Passkeys in macOS. You can save Passkeys in your Chrome profile, where they're protected by a macOS Keychain." Then under "Important" they said: "Chrome cannot save or use Passkeys stored in iCloud Keychain. If your computer is lost or your Chrome profile is deleted, you cannot recover your Passkeys."

And third: "You can use a security key to store your Passkeys. Important: Passkeys stored on security keys are not backed up. If you lose or reset the security key, you

cannot recover your passkeys." What a wonderful system. This clearly represents a huge leap forward.

Leo: Sigh.

Steve: Wow. It's clear that, unfortunately, what we have at the moment is an extremely fragile system. The problem is the extreme secrecy surrounding the private keys which create the Passkeys. It's true that they do need to be guarded. Unfortunately, at the moment they're being jealously guarded. How Microsoft could possibly imagine that it's practical to have all of a user's Passkeys locked up in a single machine, unable to synchronize with any of a user's other devices is beyond me.

But we're ready to entertain the second part of Dave's question, where he asked: "Can the passkeys be paired across two user accounts, thereby ensuring recovery in case of loss with only three keys?" He says: "My mental model said it made sense, but I do not know for sure. Can the same key be applied to two different people? To assure full backup protection, can all three keys be coded into both users?"

The answer is that not one of those operations Dave is asking for is available. Not one. And what's more, I just double-checked. As we learned last week, Yubico's YubiKeys have the most ample storage for Passkeys of any hardware Passkey dongle in the industry, and even it is limited to a total of only 25. And they are utterly and absolutely non-exportable. A YubiKey is at its heart an HSM, a hardware security module. The internal YubiKey dongle hardware contains a very high-entropy random number generator that's used to synthesize a unique private key.

That private key never leaves the device. There is no way to export it. Exportation does not exist. There's no way to put a Passkey in, and no way to take a Passkey out. This would not be a problem if sites were to allow multiple passkeys to be registered for a single account. And there's no reason that would not be possible. But how many sites today support the use and management of multiple passwords for a single account? I've never seen one. So it's unclear why support for multiple passkeys would ever be created, even though nothing prevents it.

With YubiKeys having a 25-passkey limit, other than for experimentation, they seem most practical for higher-end enterprise-grade security applications, and perhaps for eventually signing into only a few of the most secure sites where the inconvenience of having an absolute hardware-lock is warranted by its ultimate level of hardware-level security. And as we noted last week, a YubiKey might be used to unlock a password manager, which is where, we would all have to conclude, all of a user's Passkeys should probably be stored.

The only sane conclusion we can draw is that, while this is all very interesting, none of this is yet ready for prime time. Poke at it, experiment with it, but wait until Bitwarden's Passkey-supporting mobile clients emerge from their current beta-testing state, at which point it will be practical to start depending upon Passkeys because they will be in a single, sane, multiplatform client. And Bitwarden, which is we should say a sponsor of the TWiT Network, will likely be offering backup and support and exportation of those once the security protocol for doing that, which is reportedly underway within the FIDO Group, is concluded.

So Bitwarden will then generate and hold our Passkeys, even when other Passkey clients on iOS and Android might be trying to. And then of course, as we said last week, the challenge is making sure that your chosen Passkey authenticator is universally used, even in an environment where multiple authenticators are all vying for attention. So I

have to say it's the things, reasonable things that people would want to do are not available. They cannot be done.

Leo: God. Wow.

Steve: Yeah. Yeah.

Leo: Yeah. I saw there was a Hacker News story about - somebody wrote about why it's a hundred times harder to implement Passkeys on your website than you might imagine. I mean, it's just - I think this is going to be - I feel like people are going to throw up their hands and say, okay, fine, never mind. And that's depressing.

Steve: Right. Right. And as we said last week, if it doesn't achieve critical mass, then it'll just be, well, exactly as one of our listeners said - or no, no, it was the guy who did the Rust WebAuthn client. He said, "I feel that this will, you know, it'll be like ad blockers. A small percentage of people take the trouble to do it, but it's sort of a niche, and it never really becomes a problem for ad companies." And in this case it just never takes hold.

Leo: So speaking of...

Steve: I mean, it is a mess.

Leo: It is a mess. And it's not getting any better. This did not solve it. We've been trying. I mean, I remember when Microsoft tried the single sign-on thing 20 years ago. We've been trying to solve this.

Steve: And they had something called Passports.

Leo: That's what I was talking about, Passport, exactly. It was a single sign-on. And it didn't get adopted, and that's that. And...

Steve: Yup.

Leo: Oh, well. Oh, well.

Steve: So one last piece of feedback from Willie Scott.

Leo: Before you do that, can I ask the other question about assembly language?

Steve: Yeah. Yeah, yeah, yeah. Yeah.

Leo: I was thinking the other day about how one debugs in a higher level language. You'll write a print statement, for instance, and it'll tell you all your stuff. You must have some macros you've written over the years to help you debug assembly. Or do you?

Steve: No.

Leo: I knew it. I knew it. You just write it right the first time.

Steve: Well, so for - not for debugging. But, for example, one of the reasons it would be difficult for me to share my assembler is that I have built up a macro archive of things I do.

Leo: Oh, I'm sure you have; right.

Steve: For example, I use a macro called "zero," and it takes a register name. Well, it simply expands to XOR register comma register.

Leo: Right, to zero it out.

Steve: Because you know when you XOR something, exactly, when you XOR something with itself you get zeroes, and it's very fast because it doesn't depend upon a memory fetch or the previous data or the previous contents of the register. So, and the point is, if I wrote XOR something comma something, I would have to look at it and say, okay, XOR, and then look at what am I doing, and then realize, oh, I'm wanting to zero that. Well, it's much better if I just say zero and then the thing.

Leo: Right. Right.

Steve: So anyway, and you cannot do that for variables, that is, the Intel architecture will not allow you to XOR memory with another memory. You can only XOR a register with a register or a register with memory, but not memory with memory. So when I have a variable, I use the macro "reset," which moves a zero into it.

Leo: But you don't have any macros for kind of displaying the contents of the stack, purely for debugging? You don't have anything like that? You just look at the code and figure out what's going on?

Steve: Oh, no, no. So I definitely have a debugger.

Leo: Oh, good, okay.

Steve: Oh, yeah, yeah, yeah.

Leo: MASM comes with a debugger; right? Or no?

Steve: So MASM doesn't, but there were back in the day a bunch of third-party debuggers.

Leo: Right.

Steve: I use something called Periscope, which was written by a guy named Brett Salter years ago.

Leo: I remember that, yeah.

Steve: He passed away a few years ago. There was also something called SoftICE. And ICE stands for In Circuit Emulator. And in the really old days you would pull the processor off the motherboard and plug in this paddle that then had a cable running to a bunch of things that emulated the processor that allowed you essentially to get inside the processor.

Leo: Wow. That's wild.

Steve: So that was called an ICE, an in-circuit emulator. And so SoftICE was essentially using protected mode to do all the same sorts of things. So there have absolutely always been debuggers. And one of the banes of developing for SpinRite was that I'm DOS and 16 bits. And it was very difficult to create an environment where I was able to have networking in order for my code to get down into the target machine and debugging at the same time. So one of the things I'm really looking forward to as I move to my own environment is, for example, this RTOS32 that will be the home for SpinRite 7. It works with Visual Studio transparently. So I get to just live in a really nice GUI IDE and do all of my debugging.

Leo: Oh, that's nice.

Steve: And what's really cool, Leo, I bought so many motherboards and so many random hard drives through eBay when our testers were reporting that on my Gimcrack 27Z it does such-and-such.

Leo: Right, right.

Steve: And it's like, oh, my god. So I'd have to go look. I'd go to eBay, search for a Gimcrack 27E and...

Leo: And buy one.

Steve: There's one, yeah, and I would buy one. And my amazing wife put up with having motherboards everywhere.

Leo: All over the dining room table, I'm sure.

Steve: So what's very cool about RTOS32 is it allows Internet, trans-Internet debugging.

Leo: Oh, nice.

Steve: So if something is happening on that guy's Gimcrack 27Z, I'll be able to actually have him contact me and debug it on his machine.

Leo: Wow. Oh, that's really cool. Wow.

Steve: Yeah.

Leo: Very neat. All right. Okay. So you have some pretty good tools, it sounds like.

Steve: Oh, yeah. And in fact one of the things that I've learned is invest in your tooling infrastructure before you do anything. It is so nice...

Leo: Absolutely.

Steve: ...to have a convenient debugging environment.

Leo: Absolutely. On we go. I'm sorry, I didn't mean to interrupt. I was just curious. I was debugging the other night, and I was thinking, I wonder how Steve does this. So now I know.

Steve: Yeah. You absolutely have to have a good debugger.

Leo: Oh, yeah, absolutely.

Steve: That allows you to see the stack and the contents of the registers and what's in memory and what your local variables are. All of that is made really very nice with Visual Studio.

Leo: Nice.

Steve: Okay. Willie Scott. He says, okay, he has some feedback and advice about the operation of the iCloud keychain. And I bet you he knows what's going on. Or at least

gave us enough of a clue. He said: "Hi, Steve. In regards to your discussion of Passkeys on last week's show, the part about the author's partner losing her iCloud Keychain passwords intrigued me. After the LastPass hack, I decided to switch to using iCloud Keychain for my passwords because I'm in the Apple ecosystem and wanted to start using Passkeys instead of passwords wherever possible.

"I'm writing to mention that I, too, have had passwords and two-factor authentication codes wiped from my iCloud Keychain..."

Leo: Oy.

Steve: Uh-huh, exactly, "...although my Keychain has never been fully wiped, like the poor partner's Keychain did. As near as I can tell, I believe I know the culprit of why it may be wiping credentials from iCloud Keychain and wanted to pass this along to anyone who might still be using iCloud Keychain to store their passwords, or who knows somebody who may.

"When I started changing all my passwords and adding accounts into iCloud Keychain, I noticed that an old Amazon password that I don't use anymore was already stored in there, probably from when the Amazon app asked, 'Do you want me to remember your password?' It was an old password that I don't use anymore, so I deleted it. However, a couple of days later, I noticed that even though I deleted that password, or so I thought, it had somehow reappeared in my iCloud Keychain. Not only that, but I also noticed that one or two accounts that I had recently added to the Keychain were missing. And this process repeated itself a few more times. So that's when I started investigating.

"While digging through the settings, I went through my Apple ID account settings, and that's when I realized that my old iPhone 6S Plus, which was running an old version of iOS - iOS 14 to be exact - was still signed into my iCloud account and had iCloud Keychain turned on. I removed that old iPhone from my iCloud account. And ever since I did that, no passwords have been wiped. If you're in an Apple ecosystem, it's always a good idea to keep your devices up to date, but it might also be a good idea to do some spring cleaning and remove old Apple devices from your iCloud that you don't use anymore.

"Having said all that, I sadly was agreeing with a lot of the points you were making about Passkeys. And I think I've decided that I will probably switch over to Bitwarden once Passkeys become officially supported in Bitwarden, using," and he says, "https://bitwarden.com/twit, of course."

Leo: Thank you. That's our special sponsor link, yeah.

Steve: Which I think we're about to talk about.

Leo: Yes, actually.

Steve: "Thank you for a great show. I look forward to it each week. I'm also a proud SpinRite owner and can't wait to start using 6.1 on my SSDs and a troubled hard drive."

So this mysterious iCloud credential removal has all the feel of something Apple would be deliberately doing out of their typical abundance of caution. I'll bet there's a security

model behind it. For example, while an older iPhone is also signed into an account's iCloud Keychain, Apple might be deliberately limiting what they're willing to save into that shared Keychain while an older and presumably lower-security device also shares access. In other words, it's a feature, not a bug.

Leo: I guess it could be that. I don't like that kind of unexplained behavior, however.

Steve: It sounds like Apple, though, to say, oh, we're not going to let you hurt yourself. We're going to delete the keys you've just saved because otherwise one of your insecure devices might get them.

Leo: Ay ay ay.

Steve: Yeah.

Leo: I'll be sure to - you always should remove old devices. That's maybe why I've never run into this. I always remove the old devices. So, hmm. Very interesting.

Steve: Yup. I do happen to have an iPhone 6 right here.

Leo: Wow, look at that.

Steve: That doesn't work anymore.

Leo: Look at that home button and think fondly on it because Apple has, as of today, discontinued all the devices that had home buttons. The last one you could buy was the iPad base model, and that's now been superseded. So the home button is officially a thing of the past, as is the headphone jack, I think. I think...

Steve: Is it all facial recognition?

Leo: Yeah. It's all Face ID now.

Steve: Makes sense. Today's podcast is titled "Not So Fast" because that's the absolutely best way to characterize what's going on in the United Kingdom with Google. As we know, during our podcast two weeks ago Leo dropped the news that Google's third-party cookie deprecation would not be happening as had been long planned for this summer. And of course I was getting all excited about that because, you know, I've been on this third-party cookie thing for a long time. I think it was in 2008 I created that whole cookie forensics facility. GRC understands which types of assets carry cookies and which ones are first-party and third-party and everything.

I mean, and there were, back then, browsers were not handling cookies correctly. When you turned them off, sometimes they didn't get turned off. Or turning them off would keep new ones from being stored, but would not cause old ones to start getting blocked.

And there was just all kinds of screwy things that were going on. So this has been a hobbyhorse of mine for decades.

So it is the case that the abandonment and deliberate blocking of all third-party cookies and other web-tracking hacks represents such a dramatic sea change for the web that, I get it, many understandably skeptical observers doubt it can or ever will actually come to pass. And, you know, we've been abused for so long it's difficult to imagine that could ever end. So, self-confessed technology fanboy that I am, I wanted to determine what was going on. Were some stuffed-shirt bureaucrats somewhere going to screw this all up?

When I went to take a look at that for last week's podcast I quickly became lost in a paper shuffle. I decided that whatever was going on was worthy of understanding, since I consider this single forthcoming change, that the largest browser maker in the world by far wants to make, to be one of the most important things that's going on today. That and the question about, you know, are we going to keep our conversations encrypted in messaging apps, which the EU seems determined to say no to. As I've previously said, this represents a complete - what Google is doing represents a complete reconceptualization of the way the Internet will finance itself going forward. And we could have it soon.

So the news that Leo had picked up on came in the form of an announcement that left actually more questions than it answered. On the 23rd of last month - which was, you know, Tuesday before last - on their PrivacySandbox.com site, Google posted under the headline "Update on the plan for phase-out of third-party cookies on Chrome." That's very clear.

Their brief introduction said: "The UK's Competition and Markets Authority" - known as the CMA, and we'll be using that acronym a lot here, or abbreviation - "and Google publish quarterly reports to update the ecosystem on the latest status of Privacy Sandbox for the Web. As part of Google's first-quarter 2024 report, we will include the following update" - that is, in the report - "about the timeline for phasing out third-party cookies in Chrome in the April 26th report."

Okay. So the update, very short, it simply reads: "We are providing an update on the plan for third-party cookie deprecation on Chrome." They said: "We recognize that there are ongoing challenges related to reconciling divergent feedback from the industry, regulators, and developers, and will continue to engage closely with the entire ecosystem. It's also critical that the CMA has sufficient time to review all the evidence, including results from industry tests which the CMA has asked market participants to provide by the end of June." Okay, now, that means essentially June is when third-party cookies were supposed to be ending; but, you know, things are taking longer than expected.

"Given both of these significant considerations, we will not complete third-party cookie deprecation during the second half of Q4. We remain committed to engaging closely with the CMA and ICO and we hope to conclude that process this year. Assuming we can reach an agreement, we envision proceeding with third-party cookie deprecation starting early next year." So early 2025. And then they conclude by noting: "Once published, you will be able to view both Google and the CMA's full reports." Those reports were published three days later, on April 26th. So this was on the 23rd they said this. Surprised the industry. Three days later on the 26th we got the whole story.

So the entire issue is best described by the following statement: "On 7 January 2021" - okay, so a little over three years ago. "On January 7th, 2021, the CMA commenced an investigation under Section 25 of the Act" - some, you know, UK, the equivalent of legislation to prevent monopoly misbehavior, you know, antitrust we have here in the

U.S. - "in relation to Google's Privacy Sandbox proposals. The CMA subsequently informed Google that the CMA was concerned that Google's proposals, if implemented without regulatory scrutiny and oversight, would be likely to amount to an abuse of a dominant position."

So basically a little over three years ago Google says we're going to change the way the Internet is financed. And among those things we're going to kill off third-party cookies. There's no question that people in the UK whose income and livelihoods depend upon tracking, like, you know, their data resellers, they said, whoa, whoa, whoa. We don't want third-party cookies to go away. We like third-party cookies. So UK bureaucrats, please tell Google no. Please tell Google we need those cookies.

Okay. So I don't know that for a fact. It's unclear. And it's frankly not really important to know the genesis of the inquiry, but it's probably something like that. Since we're talking about the elimination of all third-party cookies and the curtailment of what had become the widespread practice of tracking Internet users around the web as a means of determining their interests, it may well have been the advertising technology companies based in the UK which were crying foul behind the scenes.

Leo: That's even more exciting, really, yeah.

Steve: Yes, yes. What ensued was about what you'd expect from any healthy and well-established bureaucracy as old and wizened as the United Kingdom. Experts were - yeah. Experts, you know, I mean, even the name United Kingdom sort of suggests, oh, crap.

Leo: [Expostulating]

Steve: Experts were found, neutral third-party "monitors" were enlisted, and Google created a document describing the - and, boy, are you going to hear this word - "the Commitments it was prepared to make," with a capital C. I mean, it sounds religious almost. These are our commitments. A document titled "Investigation into Google's 'Privacy Sandbox' browser changes" opens with the assertion that: "The CMA has accepted commitments offered by Google that address the CMA's competition concerns resulting from investigating Google's proposals to remove third-party cookies and other functionalities from its Chrome browser." Which begs the question, what exactly are these commitments that the CMA has accepted?

I found the points of concern in the description of the roles of the appointed technical expert that will be supporting the monitoring agent. The document states: "On the 26th of September 2022, the CMA approved the appointment of S-RM Intelligence and Risk Consulting Limited by the Monitoring Trustee (ING Bank N.V.) as an independent Technical Expert to support the Monitoring Trustee in monitoring compliance with the following provisions of the binding commitments accepted by the CMA on February 11th, 2022." Whew. Okay. And then the good news is this next line is short.

"Google's use of data (paragraphs 25 through 27), non-discrimination (paragraphs 30 and 31), and, with respect to those provisions, anti-circumvention (paragraph 33), the role of the Technical Expert is to provide specialized knowledge to support the Monitoring Trustee, particularly in relation to monitoring data flows and understanding the possible impacts of the Privacy Sandbox changes on ad tech markets."

Okay. So we have the ING Bank serving as the neutral monitor, and this monitor has appointed another firm with the required technical expertise. And everything it focused

upon is in a small handful of paragraphs somewhere. I found out where. They are in Appendix 1A of the latest version of the "Google's final commitments" document. The first set of paragraphs, 25 through 27, basically amount to Google promising not to use any personal data from a user's past Chrome browsing history, a customer's Google Analytics account, or to in any way track users. So that's all pretty much what Google has explained to be its intentions and goals. So it appears that the CMA just wanted that very clearly and succinctly spelled out.

The non-discrimination, that's paragraphs 30 and 31, state that Google promises to create a totally level playing field. Having examined, explored, and shared on this podcast the operation of Google's cookie-replacement technologies as they have evolved through the years, this was, you know, it was always clear to me and those who understood this that this was inherently level, the playing field was. That is, you know, Google was getting a very proscribed amount of information, and everybody was equally - had equal access to it. It's implicit throughout Google's design, though I have to agree that Google's design has grown to be much better thanks to all the feedback and criticism the various pieces have received through the years. So yes, it's a good thing we did not get stuck with Google's first idea. What we've got is something far better than what we would have had if, you know, if there was sufficient scrutiny done. And there was.

So I can understand how bureaucrats, who will never understand how Google's Topics API functions, need a simple "okay, but what does it mean" spelled out in English. Since this is crucial to the acceptance of Google's technology, and it's only two paragraphs, I'm going to share them.

Paragraph 30 says: "Google will design, develop, and implement the Privacy Sandbox proposals in a manner that is consistent with the Purpose of the Commitments and take account of the Development and Implementation Criteria. Google will ensure that it does not distort competition by discriminating against rivals in favor of Google's advertising products and services. In particular, Google will not" - and we have three things - "design and develop the Privacy Sandbox proposals in ways that will distort competition by self-preferencing Google's advertising products and services; also will not implement the Privacy Sandbox in ways that will distort competition by self-preferencing Google's advertising products and services; and, finally, also will not use competitively sensitive information provided by an ad tech provider or publisher to Chrome for a purpose other than that for which it was provided."

Then it says: "For the avoidance of doubt, Privacy Sandbox proposals that deprecate Chrome functionality will remove such functionality for Google's own advertising products and services, as well as for those of other market participants." That was paragraph 30. And yes, I mean, that's exactly what Google has said they're going to do. But essentially what has happened is a legally binding contract has been created that Google - that's what these commitments are which Google is saying they're going to honor.

And paragraph 31 just says: "Google will not change its policies for customers of Google Ad Manager, Campaign Manager 360, Display & Video 360, or Search Ads 360 to introduce new provisions restricting a customer's use of Non-Google Technologies before the Removal of Third-Party Cookies, unless in exceptional circumstances - such circumstances to be discussed with the CMA - or as required by law. For the duration of the Commitments, Google will inform the CMA ahead of any such change to these policies."

And this leaves us with the final "anti-circumvention" paragraph 33 which is just a blessedly single line which reads: "Alphabet Inc., Google UK Limited, and Google LLC will not in any way, whether by acts or omissions, directly or indirectly, circumvent any of the Commitments." Now, that sort of language will be familiar to any businessman or anyone

who's been involved in any contractual agreements where attorneys are engaged. You know, it's boilerplate; right? And it's important to understand that both the United Kingdom government and Google's various corporations recognize those provisions to be now contractually and legally binding.

So it has been upon those representations, which are enumerated as "Commitments" with a capital "C," that the UK then proceeded to carefully examine Google's proposal. So now we return to the timeline for phasing out third-party cookies. That work appears in a document titled "CMA Q1 2024 update report on implementation of the Privacy Sandbox commitment," dated last month, April of 2024. Actually it was April 26th.

The document's summary lays out the entire story, and it's interesting enough and short enough to share. They said: "This report sets out the CMA's updated views on the issues we identified in our January 2024 report." So January was the previous report. So it's basically quarterly; right? So this is the result of the first quarter. So this is from January 2024. Where are we now? We're in April. So we've had the first quarter go by. "Our analysis is based on the framework for assessment set out in the legally binding Commitments that Google made in February 2022 to address competition concerns relating to its proposals to remove third-party cookies from Chrome." So in other words, yeah, this is a big deal for the entire Internet. It's a big deal. "The January 2024 report set out our provisional views on the impact of the Privacy Sandbox on competition, publishers, and advertisers, and user experience.

"We outline Google's response to the concerns we identified in that report, the January report, and the steps it is taking to resolve pending issues. We've also considered the feedback received from market participants on these points. We've included a summary of this feedback in the sections below.

"This report also incorporates the preliminary assessment of the ICO, the Information Commissioner's Office, on the privacy and data protection impacts of the Privacy Sandbox. Having consulted with the ICO, we set out our current views on these concerns for each of the APIs.

"Although there are a number of concerns to work through, based on the available evidence, we consider that from 1st of January 2024 through the 31st of March 2024, the relevant reporting period, Google has complied with the Commitments. This means that, in our view, Google has followed the required process set out in the Commitments and is engaging with us and the ICO to resolve our remaining concerns ahead of third-party cookie deprecation.

However, further progress is needed by Google to resolve our competition concerns ahead of deprecation. We will continue to work with Google to resolve our concerns between now and the point at which Google triggers the Standstill Period. We will provide an update on progress in our next update report. Testing of the Privacy Sandbox tools is also currently underway. The test results will form part of a wider evidence base that we will use to assess the effectiveness of the Privacy Sandbox. The test period runs until the end of June this year." And as I said before, because this is running through June, that's what kept the cookies from being, you know, for the beginning of the deprecation to start at the end of June.

They said: "Given the time needed to resolve outstanding issues and take account of testing results, we have agreed with Google that there should be a limited delay to third-party cookie deprecation. Subject to resolving our remaining competition concerns, Google is now aiming to proceed with third-party cookie deprecation starting in early 2025. Under the Commitments, it is for Google to decide when the Standstill Period is triggered. We encourage market participants taking part in testing to submit their results directly to us by the end of June deadline. We also welcome any additional feedback from

stakeholders on the concerns identified in this report. Our contact details are included at the end of the report."

Okay. So one last thing. This made reference to a "Standstill Period" several times, so I tracked that down. In the earlier Commitments documents it appears to be just more bureaucracy for its own sake. It says, on paragraph 19: "Google will not implement the Removal of Third-Party Cookies before the expiration of a standstill period of no less than 60 days after Google notifies the CMA of its intention to implement their Removal. Google may increase the length of such a standstill period at any time between giving such notice and the period's expiration. At the CMA's request, Google will increase the length of this standstill period by a further 60 days to a total of 120 days."

Okay. So what follows all of that is - that was the document summary. There are 97 pages of interesting, but ultimately mind-numbing, back and forth detail, as every conceivable facet of this big change Chrome will be implementing is examined under a bureaucratic microscope. The real concern is over Google's size and whether the changes it is making will disadvantage smaller ad tech players. But what becomes clear after reading at least some, and that's what I did, I could not go through 97 pages of this, my eyes started to cross and I couldn't see, it is very clear that the UK is moving clearly in Google's direction.

Both parties are truly negotiating in good faith. That's one thing that also is very clear. This is not the UK stonewalling and being unreasonable. It really is, as Leo portrayed, a bureaucratic walrus that just absolutely does not have any idea what is going on. People are nipping at it, saying this is bad, you can't let Google do this. So Google is saying this is not bad, this has to happen, we want to stop tracking on the Internet. People who make their living from tracking are saying, yeah, but we like tracking.

Leo: Yeah.

Steve: Yes. And so the UK is sort of stuck in the middle. Google is being reasonable. They are, I mean, there must be a division of Google where they're intoxicated in hot tubs somewhere, just in order to maintain their sanity. There's no way that the developers are dealing with any of this nonsense because, I mean, ultimately that's what it is. But the UK needs to be placated through having this explained, you know, what exactly this is and does. So that's what's happening. Again, progress is being made. In the January Report, for example, there was an instance where the ad tech companies were trying to claim that because of their shorter reach, they were being disadvantaged. The expert looked at it under the watchful eye of the monitor, and now in the April Report the conclusion is, no, that is not the case. There is no disadvantageous handling based on size of advertiser. We see no evidence of that. We understand the technology. That's not the case.

So it does not appear to me that Google's Privacy Sandbox technology is in any trouble at all. The truth is, as I've said, it represents a massive change to the way the Internet pays for itself and is going to fund itself in the future. And it's also true that many companies whose revenue has been entirely derived from the oh-so-slimy practice of tracking users and aggregating their data without our knowledge or permission, for the purpose of selling that data to anybody with a wallet, will be - their income will be impacted. And not in a good way.

So having read through the documents, I can understand that the process is taking place, and it's taking time. And in retrospect, you know, though I would have never expected this would happen, it is at least understandable, and it appears that the world will indeed soon be receiving this dramatic change in the way Internet-based advertising

is carried out. It is, you know, clearly far superior to the status quo. I mean, we can't keep going on the way we have been. And it takes something no less large than Google to just simply make it an ultimatum. We are going to do this. So I understand they've got to satisfy the walruses of the world. It looks like that process is close to being done.

Leo: Yeah. I hope so. Of course advertisers don't like it. That's why we like it. And I think Google, obviously they're trying to balance the interests of both parties because they are - they sell ads. They buy ads. It's their business. It's their revenue. But they also understand that consumers are not happy, and I think they need to...

Steve: No.

Leo: ...find a way everybody's happy.

Steve: And Leo, I'm impressed by the minimization of the information that Google themselves are willing to obtain. I mean, it's, as we've seen, Topics is not invasive. They are, you know, no one can be identified from their Topics. They are chosen at random. I mean, the system has incredible checks and balances built in which we've talked about on the podcast when we explained it. And I think we're probably due for a re-explanation when it actually goes into effect because, you know, it's the way the world's going to work. And I loved the comment about the reason my machine's fans were spinning up was that my Chrome browser, when I was running Chrome, was busy holding auctions with all of the world's ad agencies.

Leo: Yeah, well, that's coming, maybe, anyway. We'll see.

Steve: It's the only way to do this is to make it user side.

Leo: Yeah.

Steve: You move it to the user, and then the user's browser chooses what they're going to see. It's brilliant.

Leo: Yeah. Makes sense.

Steve: I'm going to tease next week's topic, I believe. I think next week's topic will be ZTDNS, which stands for Zero Trust DNS. Last Thursday Microsoft published a preview of a forthcoming security solution they call "Zero Trust DNS." It's been clear for a long time that DNS represents, as we know, both an Achilles heel of network security and a point where it's also very possible, if you're clever, to introduce a significant new level of security. From my brief scan of the technology Microsoft has outlined, it appears that any of our listeners who may have followed up on my discovery a few months back of ADAMnetworks' DNS solution, which they call "Don't Talk to Strangers," may already be enjoying the benefits of dramatically improved security, thanks to leveraging the power of DNS. But I needed more time to dig into what Microsoft is doing. So for next week's podcast I plan to take a deep look into what Microsoft has announced.

Now, one thing I should say that immediately stood out was that Microsoft might be attempting to use this as a way of driving enterprises to Windows 11, since enterprises don't want Windows 11, as we've heard Paul Thurrott mention many times. No one really does. And in Microsoft's diagrams which I briefly scanned, they're explicitly labeling the clients as Windows 11 machines.

Now, that just might be Microsoft, you know, because Windows 11 is what they're all using. You know, since no one actually wants Windows 11, since Windows 10 still commands more than twice the number of desktops as Windows 11, and a much greater percentage within the enterprise because most, you know, new computers come with Windows 11, but enterprise machines that have been running for 10 years don't. And since a huge install base of machines won't even run Windows 11, if what Microsoft is planning to do is truly a Windows 11-only solution, then the client agnostic system that the ADAMnetworks guys already have working and well-proven seems like a far more practical one to me.

But in any event, by the end of next week's podcast, we'll know exactly what's going on. And, you know, it's a good thing that Microsoft is stepping up and looking to improve DNS security because we all know it needs it. But seems to me there's already a solution in place. But not from Microsoft. And so when the biggy does it, you know, I remember, Leo, it was fantastic. Brad Silverberg and Brad Chase came down from Redmond and took me out to lunch and said, Steve, we're going to be announcing DOS 6 pretty soon, you know. And I said, uh-huh. And they said, "We're a little self-conscious about this, but we're adding something called Scandisk."

Leo: Oh. It's nice of them to warn you.

Steve: "Now, don't worry."

Leo: It won't work well.

Steve: "It's doesn't do what SpinRite does." And I said, "Uh-huh, great. That's just effin' wonderful."

Leo: Was that later than Chkdsk? Because they had Chkdsk.

Steve: Yes. For the rest of our existence we are answering the question, "Well, I already have Scandisk. What do I need SpinRite for?"

Leo: Oh, right.

Steve: Anyway, the point is, it matters when the giant offers the...

Leo: It does, yeah.

Steve: Oh, it does.

Leo: Yeah.

Steve: I've been there firsthand. I liked Silverberg a lot. I never was fond of Brad Chase.

Leo: You weren't Sherlocked, though, as the Mac people call it. So that's the good news.

Steve: No, Norton did that.

Leo: Tried. Do you think he sold more copies of Disk Doctor than you sold of SpinRite? Probably, huh.

Steve: Well, what he did was, when I refused to sell SpinRite to Peter, he sent a developer home with a copy of it and said...

Leo: Oh, that's not nice.

Steve: Oh, yeah. And we know that because one of my guys looked inside and saw code that was our code. I mean, there was a place where I needed to see whether the BIOS handled a certain API call. So I put some specific random data in the registers when I made the call to see whether they got changed.

Leo: Oh, that's kind of a smoking gun.

Steve: Their clone of SpinRite...

Leo: So the same data.

Steve: ...used the same values, the same data, because they didn't know what I was doing.

Leo: They didn't know, exactly. They said, well, we'd better do it this way because we don't know if it does something.

Steve: Yeah. The good news is, since they didn't actually create - what was it called? Calibrate was their clone. Since they didn't create it, when their customers called for support, they said, "Well, we're not sure. Call Gibson Research." I'm not kidding. We got calls for our support people, "Well, Norton said to ask you about Calibrate." And we said, "Well, when you buy a copy of SpinRite, we'd be happy to answer all your questions."

Leo: We'll tell you. Steve is at GRC and still selling SpinRite, now v6.1, many moons later. And it's even better than ever. In fact, now it speeds up SSDs.

Steve: Yup. It's doing really well, actually.

Leo: Yeah, yeah. Really, congratulations.