SECURITY NOW!

Transcript of Episode #770

Zoom's E2EE Debacle

Description: This week we take an interesting new look at some new problems arising with DoH; we look at IBM's new stance on facial image recognition research; we look at two recently disclosed flaws in the Zoom client; we check on the severity of the latest UPnP service flaw; and we update on Microsoft's new Edge rollout. We share a bit of miscellany and some terrific feedback from our listeners, touch on my SpinRite project progress, and then explore last week's truly confusing Zoom encryption reports that give the term "mixed messaging" a bad name.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-770.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-770-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We'll talk about the DoH DoS. We also talk about IBM abandoning face recognition technologies and the weird story about Zoom's encryption. Is it, or isn't it? Steve parses the statements, next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 770, recorded Tuesday, June 9th, 2020: Zoom's E2EE Debacle.

It's time for Security Now!, the show where we talk about the latest news in security. We get you some help from this guy right here. He's the security guru, the king, Steve Gibson. Hello, Steve.

Steve Gibson: Yo, Leo. Wow, 770 episodes.

Leo: No. I have 740. Oh, you're right. I've got to fix this, too. Whoops. Thank you for correcting.

Steve: Glad I happened to make a note of that, since we want the proper lower third down there.

Leo: Is that a magic number?

Steve: And this is 2020, by the way, Leo.

Leo: Oh, good. Okay. We've got the date right.

Steve: We have June 9th, 2020.

Leo: Okay. The day's right. Just the episode wrong.

Steve: So we went from last week's podcast, Zoom's E2EE Design, now to Zoom's E2EE Debacle.

Leo: Yeah, that was quick, snatching defeat from the jaws of victory, so to speak.

Steve: I cannot wait to get to the end of the podcast because what happened last week, you've just got to scratch your head and say, what? Who? What? Huh? Anyway, we'll see why. But before that we're going to take an interesting new look at some new problems that might be arising with DoH, making it much more like "doh" than we thought. Something happened last week with Firefox when it unleashed DoH onto NextDNS that wasn't happy.

We're also going to look at IBM's new stance on, and I'm happy about this, facial recognition research. We're going to look at two recently disclosed flaws in Zoom's client, which is separate from the debacle. You can have a mistake, and you have a policy. We always are careful to separate those. We're going to check on the severity of the latest, yes, once again, we're back to UPnP...

Leo: Oh, no.

Steve: ...with a flaw. Oh, yeah. And the tech press is hyperventilating again. We're also going to update on Microsoft's new version of Edge, its rollout. We've got some miscellany, just a couple grc.sc shortcuts I want to share relating to the COVID stuff and where we are. We also have some terrific feedback from our listeners. I'm going to touch on the recent progress since last week on SpinRite project. And then we're going to look at last week's truly confusing Zoom encryption reports that give the term "mixed messaging" a bad rap.

Leo: Oh, man.

Steve: Yeah. And we do have a fun Picture of the Week. So I think another great podcast.

Leo: We have that all still to come.

Steve: 770.

Leo: Yes. I fixed the number; right? It looks good now?

Steve: Okay. We're in agreement now on the episode number.

Leo: I'm glad you mentioned that.

Steve: So our Picture of the Week is not about security directly. Even, yeah, really not at all. I just liked it. It was just fun. It came to me, and I wanted to share it with our listeners.

Leo: This is the story of your life.

Steve: Yeah. Yeah, exactly. So I gave it the caption "It's been a weird year." And this is a time graph that labels itself "Relative Importance in 2020, So Far." And there are a number of items. It goes January, February, March, and April. And so one of the items is coffee. And that's a straight line across the top. We needed it before all this craziness, still need it now. The curve for car has of course dropped off dramatically. We were using our cars a lot at the beginning, not so much now. Internet goes up from where it was, up like right up near the top. And around the beginning of March the line for shaving, which was a straight line across, drops down to zero.

Leo: Plummets.

Steve: Yeah, don't need to do that so much. The real funny one, actually, is alcohol, which is moving along on a straight line until March. Then not only does it go up, but it goes off the top of the chart through the title and off the screen. So, uh-huh. And then of course there was that weirdness about toilet paper, so it represents that with a slightly trending upward line that suddenly jumps up to the top for a while, then comes back down where it was before. Sweatpants is the final parameter that is tracked, and it shows a mild increase, just sort of going up over time to where we are now. So anyway, just sort of a wacky, fun little bit of, as I said, it's been a weird year in our lives. And it's - yeah.

Leo: It's not over yet, either.

Steve: No, it's not over yet. So we start this week with a bit of browser news, the odd case of Mozilla's DoH DDoS. The Mozilla team was busy last week. Tuesday they released their brand new shiny next Firefox, Firefox 77. When we were talking last week, we were on 76, and that was brand new and shiny. But no, now we had 77. That was of course for the desktop versions, Windows, macOS, and Linux. It fixed several security issues. Five vulnerabilities received a high severity score, with three of them allowing bad actors to run arbitrary code on vulnerable installations. Not what you want in your browser, so those are gone.

As Mozilla puts it, a flaw classified as high in severity, "can be used to gather sensitive data from sites in other windows, or inject data or code into those sites, requiring no more than normal browsing activities." So those got fixed. 77 also further rolled out Mozilla's WebRender project, which we've not spoken of before. WebRender is their new and emerging RUST-written 2D graphic web renderer for their browser which can be used on any NVIDIA-equipped Windows 10 machines, currently laptops, having screens of any size. For a while it was size limited. It was less coverage. They're thinking, hey, you know, browsers are doing lots of 2D stuff. Now GPUs are just present. And I think they're targeting laptops because they're also power sensitive. And so you might as well use an application-specific IC, namely a GPU, to do this stuff.

And, now, we've talked about the insane rendering that browsers are now doing. I mean, we sort of take it for granted. But there's, like, fading effects, and things are, I mean, I'm seeing stuff where, as you scroll the page up, things kind of emerge from the mist, or they suddenly expand to full size, I mean, all that's being done browser-side. And that all requires cycles. If you're having to do that in a general purpose processor, it's having to work a lot harder than if you're able to use a processor that, for example, just automatically has the ability to do multilayer opacity masking and mapping and all that. So that's what they're doing, more of that now in 77. It's looking like this project is going to be a success.

However, shortly after the release of Firefox 77, things quickly went sideways. Taking a piece of this week's listener feedback out of sequence, because I originally had it down, because I thought it was interesting, down in listener feedback, later in the podcast. And I thought, oh, this just pops to the front now.

Chris Miller, who tweeted from @Mil_Fi, he sent: "Hello, Steve. Longtime listener of Security Now!. Just want to let you know about something. I work for a fairly large county government. We have all internal users," he says, "6,000+, go through a proxy server for security." Good. He says: "Well, our proxy was overwhelmed yesterday with anything going through it. It turned out that DoH was automatically enabled on just a few of those users who had upgraded to Firefox 77," he says in parens, "(fewer than 10), and it completely crippled us."

He says: "Firefox is not a browser that many in our environment use, either." He says: "I also read the links below and saw that DoH basically overwhelmed NextDNS, the secondary provider in Firefox. Now, Mozilla appears to be slowing down its rollout tremendously. I guess the load is so much more on existing systems. Just thought you'd like to know. I'm sure other enterprises will be experiencing a similar issue." Well, and not only did other enterprises experience it, NextDNS was effectively DDoSed by the rollout of Firefox 77, which for the first time fully enabled DoH on that browser. So indeed, Chris was right.

Leo: Were they the default? Or was Cloudflare the default? Who was the...

Steve: I'm not sure, like, what the logic was. Maybe they were choosing them randomly. But NextDNS is the second-listed provider, and it buried them.

Leo: Oh, wow.

Steve: So they immediately stopped the 77 rollout and replaced it with 77.0.1, which is what anyone who is current will now have. Okay, so what happened? Interestingly, the exact details are surprisingly thin. I was really interested, so I dug around a lot. Over on Bugzilla, which of course is Mozilla's bug tracking site, only two incomplete and somewhat cryptic explanations are found. One is "Disabled automatic selection of DNS over HTTPS providers during a test to enable wider deployment in a more controlled way." And the second is "We need to be able to roll this out gradually so that we don't overload any providers. Even the dry run involves up to seven requests per client, which can be very significant when the entire release population updates."

So here's one thing that may be going on. Web servers are not super happy with long-duration persistent connections of the type that DoH defines and requires for performance which is intended to compete with traditional UDP. And of course nobody wants less performance from their browser if they switch to DoH. Yeah, hey, if the security and privacy is free, I'll take it. But not if it slows down my pages.

So it turns out I ran across exactly this problem a few years ago at one phase of the SQRL project. We wanted a SQRL login site's web page to automagically update once the user had logged on, either optically with their phone, which would see the unique QR code on the page, or with a SQRL client installed into the same machine. And that's what it does; right? In all the demos of SQRL, it's just like, ooh, click, and you're logged in. It's magic.

So there are two ways this could be done from within the web page: Either have JavaScript on that page bring up one persistent connection back to the website by having the page connect back to the server and wait for a signal to refresh the page, or sit in a loop periodically and continuously probing the website to ask whether that page should be updated. I initially took the first approach since that seemed much cleaner; right? Set up and camp out on one connection and wait for word from the server. Also it would be quicker. You wouldn't be waiting for like the next ping to come back with an answer.

So I brought that solution online, and the gang in the GRC SQRL newsgroup began playing with it. And I quickly became aware and came to appreciate just how much web servers are designed to be inherently transactional. They want to field many short-lived connections; return the data; and, if there's nothing else, hang up the connection. Having many connections all churning is no problem, but they should be coming and going rapidly. In this case, the long-term static connections were making GRC's web server very unhappy.

Back in the early days of this podcast we talked about the old-style DoS attack, not DDoS, just DoS. And in that simple DoS (Denial of Service) attack, a single low-bandwidth attacker could bring down a beefy website simply by sending a stream of TCP SYN packets. Every incoming SYN packet was a request to establish a new TCP connection. So the server would jump to action upon receiving each SYN packet. It would allocate some resources to manage that nascent connection. It would record the sequence number. That's what SYN, the sequence, that's what SYN of SYN packet stands for because it's the client saying, when I send you things, let's start numbering the bytes with this sequence number, which is a 32-bit value.

So the server would record that sequence number provided by the remote client in its nascent connection structure. Then it would generate its own sequence number to number its own replies or transmissions and record any other connection-specific details provided by the caller. Then it would finally generate and send back its own answering SYN/ACK packet, which was acknowledging the SYN received and also sending its sequence number back to the client. And if no answering ACK was received, it would assume that the reply was lost, so it would retry that several times.

The point is all of that effort and allocated resource was forced upon the server side by someone simply and mischievously sending a single SYN packet or, practically, a stream of short and simple SYN packets. So in my case, with SQRL, everyone who was sitting at a SQRL login page had established a persistent connection to my server, and it ended up being seriously overburdened. So I changed the login page's logic to instead issue the equivalent of a TCP ping, a query for a named object that would immediately generate a reply and disconnect. And I never had another problem since because that's the way web servers are designed.

So what's interesting, I mean, no one's talking about this. I've looked everywhere. I haven't found anything further. But the potential trouble with DoH is that it, too, inherently relies upon the maintenance of a persistent static TCP/TLS connection across which occasional DNS query flurries will transit. As our web browsers begin using DoH, every single browser that's open will establish and maintain a static TCP/TLS connection back to its chosen DoH provider. So I sure hope that this has been given due consideration by those who wish to move us to DNS over TCP because it's a different ballgame.

The traditional DNS, as we know, that we've always been using, is the lightest weight query we know how to make - a single, isolated, no-connection-required UDP query packet, and a returning UDP response. So no matter what, we are heading toward a solution that is a great deal more burdensome on DNS providers than UDP has ever been. Even though that connection still only transits the equivalent of UDP DNS queries, the fact that you have that connection, you know, you're getting authentication, you're exchanging an identity certificate, you're verifying the certificate, you're bringing up encryption, there's a lot more.

But all of that means that, for every single browser open, there's a connection back to a DoH provider. And that's a different ballgame. I don't know about you, Leo. I run with one or more browsers open just like, you know, it's my portal to the world.

Leo: Oh, yeah, it's always open, yeah.

Steve: Exactly.

Leo: I think some machines it's auto start. It loads at the boot.

Steve: Yeah, yeah.

Leo: That's really interesting. And there'll be a connection for each page you go to, as well; right?

Steve: No.

Leo: So if you have a hundred tabs - oh, just one persistent connection.

Steve: Right.

Leo: Okay.

Steve: So it'd be one persistent connection. And as we know, when you open like a New York Times page, it's a mass...

Leo: It's a whale.

Steve: ...of DNS queries that are going out for all of the different sources of content on that page.

Leo: Right.

Steve: I mean, it's hundreds of different domains that are involved. And that's fine. We do that now over UDP. The problem is something happened with Firefox that basically, well, it melted down that one corporate proxy that I shared from one of our listeners. And it brought NextDNS to its knees. It DDoSed them. So it wasn't from queries. It had to have been from connections. And because that model is so different, I don't know what they're going to do. I mean, they're going to need a, you know, back then when we were talking about SYN DDoSes, there was something known as a SYN cookie. And actually Daniel Bernstein and I independently invented it. He preceded me by some years. I wasn't aware of it when I thought, there's got to be a solution to this.

And the idea was you could do a stateless TCP connection. But it's stateless only until the far end, the client, responds with a SYN/ACK or an ACK in response to your SYN, thus finishing the three-way handshake and establishing the connection. At that point, you still need - it's a resource-consuming thing to maintain a TCP connection. So again, it may mean that the DoH providers are going to have to come up with much beefier servers that are able to - we're talking, what, millions of connections. I mean, how many browsers are there that could be open at the same time? That's a lot.

And so it's way different than just UDP, little UDP packets whisking in and out as people change pages. So you still have that now over TCP, but that TCP is always up. And it has to be because you can't afford the connection setup every time you refresh a page or change tabs or bring up a new page. The point is you establish that static connection, and it persists. And it looks like it's bringing down our DoH providers.

Leo: Maybe this wasn't such a good plan.

Steve: Right.

Leo: Seemed like a good idea at the time.

Steve: Yeah, it did. You know, it always struck me as a little bit kind of homebrewed.

Leo: Little janky, yeah.

Steve: Like oh, yeah. Like there ought to have been a better way to encrypt the DNS queries, rather than just basically set up a VPN is the equivalent of what you're doing. Every browser now has a VPN connection because that's what it is. Although even VPNs are smart enough to do that over UDP because it's better.

Leo: Right.

Steve: So anyway, we have a nice bit of news from IBM's CEO, Arvind Krishna. He sent a letter to Congress via Axios and CNBC, which was an odd path for it to take, but I guess he thought he'd get a little bit of PR for IBM in the process. It stated that the company, IBM, has willfully exited its general purpose facial recognition business. I'm quoting the letter in the show notes.

He said: "IBM no longer offers general purpose IBM facial recognition or analysis software. IBM firmly opposes and will not condone uses of any technology, including facial recognition technology offered by other vendors, for mass surveillance, racial profiling, violations of basic human rights and freedoms, or any purpose which is not consistent with our values and principles of trust and transparency. We believe now is the time to begin a national dialogue on whether and how facial recognition technology should be employed by domestic law enforcement agencies."

He continues: "Artificial Intelligence is a powerful tool that can help law enforcement keep citizens safe. But vendors and users of AI systems have a shared responsibility to ensure that AI is tested for bias, particularity when used in law enforcement, and that such bias testing is audited and reported." He says: "Finally, national policy also should encourage and advance uses of technology that bring greater transparency and accountability to policing, such as body cameras and modern data analytics techniques."

So that's good. CNBC noted that it's relatively easy for IBM to back out when facial recognition wasn't a major contributor to its bottom line. But, you know, certainly the media buzz may be as important as anything. IBM is still a major company, and it frequently works with governments. So this could spur some providers to follow suit, and might even get some would-be customers to drop facial recognition entirely. We'll have to see how this goes.

And of course, while some facial recognition systems may only correlate faces with publicly available data, that's what what's-their-face AI, I'm blanking on the name now [Clearview AI], the one we've been talking about all the time, you know, they're only doing public scraping of stuff; but still we've seen that it's not what people expect. It's not sort of the inherent sense of privacy from sharing a photo with friends on social media. You don't expect it to be hoovered up into a database and cross-referenced and have all the people's faces in it identified and tagged and associated in creating a big interlinked web of who knows who.

So anyway, I'm delighted with the attention that this topic is generating. We do need to talk about this. As we know, and we've said before, just because we can do something doesn't always mean that we should.

Meanwhile, I'm always amazed at companies that are looking at each other's products. And I'm glad for it. I'm glad that the security companies are taking it upon themselves to poke into other things and then responsibly disclose them. In this case, Cisco's Talos group found two critical flaws in the Zoom client. And that's interesting to me because Cisco is a big Webex provider. Anytime I'm seeing video of talking heads on CNN, they have got some sort of deal with Cisco's Webex because it advertises and brands all of their video conferences as Cisco Webex.

Anyway, we'll be talking, as we know, later about last week's mega Zoom policy issues at the end of the podcast. But in the meantime there was a problem with the latest Zoom client such that you're going to be wanting to make sure you're running the latest one. The researchers in Cisco's Talos group revealed last Wednesday that they had discovered two critical vulnerabilities in Zoom's software that could have allowed attackers, and could still if you haven't updated, to remotely hack into the computers being used by group chat participants or an individual recipient. Both flaws are - wait for it - path traversal vulnerabilities. One of the others we keep...

Leo: Not again.

Steve: Can you say dot dot backslash dot dot backslash dot dot. And even forward slash, as we learned a couple weeks ago. Yes, path traversal vulnerabilities that can be exploited to write or plant arbitrary files on the systems running vulnerable versions of Zoom video conferencing, allowing it to execute malicious code. According to the researchers, successful exploitation of either of the two flaws requires either none or very little interaction with targeted chat participants, and can be executed simply by sending a specially crafted message through the chat feature to an individual or a group. So essentially, it was a built-in remote nuke.

The first vulnerability, CVE-2020-6109, resided in the way Zoom leverages the GIPHY service which was recently acquired by Facebook to allow its users to search and exchange animated GIFs while chatting. Researchers found that Zoom was not checking whether or not a shared GIF was loaded from the GIPHY service. This allowed an attacker to embed GIFs from a third-party attacker-controlled server, which are then cached and stored on the recipient's system in a specific folder assigned with the application. But since the Zoom app was not sanitizing the filenames, attackers could perform directory traversal to save the malicious files disguised as GIFs to any location on the victim's system, for example in the startup folder, where they would then get executed next time the person started up their machine.

Leo: Wow.

Steve: So good thing that's fixed. The second remote code execution vulnerability, and that's 6110, resided in the way vulnerable versions of the Zoom application process code snippets shared through chat. Cisco wrote, and this is interesting: "Zoom's chat functionality is built on top of the XMPP standard, with additional extensions to support the rich user experience." Unfortunately, it was a little richer than they had intended.

"One of those extensions supports a feature of including source code snippets that have full syntax highlighting. The feature to send code snippets requires the installation of an additional plugin, but receiving them does not. This feature is implemented as an extension of file sharing support," wrote Cisco. So this feature creates a zip archive of the shared code snippet before sending, and then automatically unzips it on the recipients system.

And according to the researchers, Zoom's zip file extractor does not validate the contents of the zip archive before extracting it. This allows the attacker to plant arbitrary binaries onto targeted computers. Cisco wrote: "Additionally, a partial path traversal issue allows the specially crafted zip file to write files outside of the intended randomly generated directory," thus bypassing a mitigation that Zoom had put into place, rendering it ineffective.

The researchers tested both flaws on version 4.6.10 of the Zoom client app and responsibly reported it to the company. Last month, Zoom patched both critical vulnerabilities with their release of version 4.6.12 for Windows, macOS, and Linux. And as we have observed before, anyone can make a mistake. Those were that. They quickly fixed them upon being notified. While we might wish for perfect software, no one is willing to pay what perfect software would cost. So instead, we muddle through with software that mostly works and fix problems as they come to light. It's not a perfect system, but it's the one we have.

So yesterday, on June 8th, the Internet's tech press blew up over the disclosure of yet another newly discovered vulnerability in UPnP. Big surprise. Headlines took the form, for example, ZDNet: "CallStranger vulnerability lets attackers bypass security systems." BleepingComputer's headline: "CallStranger UPnP bug allows data theft, DDoS attacks, and LAN scans." And Tenable had the longest headline: "Universal Plug and Play (UPnP), a ubiquitous protocol used by billions of devices, may be vulnerable to data exfiltration and reflected amplified TCP distributed denial of service attacks." Whoa.

Since our listeners may be encountering these dire and breathless warnings in coming days, I wanted to take a moment to explain what's going on and what isn't, especially because most of the coverage is as unclear as the vulnerability's website, which naturally and predictably makes quite a big deal about it. It's CallStranger.com. So remember that the early problems with UPnP arose from the fact that a sample implementation code made available by Intel back in the year 2000 - 20 years ago, two decades ago - which was quite clearly marked "Sample" was never intended to be implemented in the field. But Intel posted a sample of here's how you do UPnP. So many, if not all of the vendors back then...

Leo: Of course.

Steve: ...grabbed that source code.

Leo: Why [crosstalk] yourself when you can just [crosstalk].

Steve: That's right. You could just, you know, you didn't even have to comment out "Sample" because it was already commented. They compiled it for their chipsets, added it to their routers, and then added a UPnP-compatible bullet point to their router's feature checklist. And so during the 20 years since, we've had many problems that are directly attributable to that original publication of never-claimed-to-be-ready-for-use code. In fact, our own episode 389, I think it was 2016, or maybe earlier than that, it resulted in me adding a public UPnP exposure test to GRC's ShieldsUP! facility. And since then, I checked this morning, 54,954 tests have come back positive for public UPnP exposure.

Okay, now, remember, it's never supposed to be publicly exposed. The idea was that it was a way for things that wanted to be discoverable on your LAN, like famously an Xbox, to solve the problem, well, such as it is a problem, of NAT routers being wonderful firewalls. You know, they're wonderful firewalls. They will not accept incoming unsolicited traffic except unless you have an Xbox and you want, you need to be able to accept incoming unsolicited traffic in order to participate in whatever it is that Xbox gamers participate in. Some network of some kind.

So in that case, the NAT router is a problem. We're going to solve that by creating a new protocol that runs on everybody's router called Universal Plug and Play. It's got nothing to do with plug and play, which allowed you to plug in a USB into your Windows machine and have it be recognized, or plug in other things and be like, oh, look, there it is. Something just appeared. Let's talk to it. No. They called it UPnP due to an apparently critical shortage of imagination when they were trying to name this thing. So this is a LAN-side server that allows anything on your LAN to say, basically, defeat the firewall which your NAT router is inherently to allow incoming unsolicited stuff on some port to go to that device that has said "Send me your unwashed packets that arrive at this port, and I want them."

So the problem is there's no security because you can't have any security because then it wouldn't be automatic. It wouldn't be Universal Plug and Play, it'd be, oh, one more thing I have to configure. And how do I tell my light switch how to receive packets from China? Well, you may not want it to; now it can. And of course the standing advice on this podcast has long been "Turn it off, everybody. Just say no to Universal Plug and Play."

To make matters worse, some really ill-begotten UPnP services on routers decide, let's not just restrict this to the LAN. Let's open it up to the WAN. Let's let everybody have this service from this IP. Oh, lord. And as I noted, 54,954 people who thought, huh, I wonder if I'm doing that. They went to ShieldsUP!. They clicked the instant UPnP test. Oh, yes, sure enough, they've got an open port 1900 on their IP. Oh, well. This is the world we're in today.

Okay. So the good news is this current CallStranger bug is not, maybe not a public exposure problem, except maybe it is. Nothing is clear. And so UPnP is not supposed to be publicly exposed. And we also know that it's not just our router that offers UPnP. Lots of printers do, and other things like cameras. Anything that wants to offer a service, this is sort of the universal, "Hi, I'm here to serve stuff." So remember SSDP is Simple Service Discovery Protocol. SSDP is what UPnP offers, Simple Service Discovery Protocol.

So this CallStranger thing is a problem which has been discovered in most current UPnP implementations. Windows 10 has the problem. And that means almost certainly all Windows versions including all of Windows servers. That's the upnphost.dll. Xbox One has the problem. Devices from ASUS, Belkin, Broadcom, Canon, Cisco, D-Link, Epson, HP, Huawei, NEC, Philips, Samsung, TP-Link, TrendNet, and Zyxel. And those are just the ones that have been tested so far. There are doubtless others. The good news is there is at least one known UPnP stack, "miniupnp," which after 2011 has not been vulnerable. So not absolutely everything is. But apparently lots are.

CERT's title for this is probably the most sane. They said: "Universal Plug and Play (UPnP) SUBSCRIBE" - in all caps because that's the verb used - "can be abused to send traffic to arbitrary destinations." Their vulnerability disclosure says: "A vulnerability in the UPnP SUBSCRIBE capability permits an attacker to send large amounts of data to arbitrary destinations accessible over the Internet, which could lead to a Distributed Denial of Service, data exfiltration, and other unexpected network behavior. The OCF" - which is the organization that maintains the UPnP spec. "The OCF has updated the UPnP specification to address this issue. This vulnerability has been assigned CVE-2020-12695," because that's just how the year's gone so far, 12,695 CVEs, and we're not halfway through. And they said: "This is also known as CallStranger."

Okay. So there are two problems. By far the largest problem, because it affects potentially all UPnP hosting devices on a LAN, and remember as I said that would include our routers and printers and probably anything similar that offers services, so perhaps even LAN-attached IP cameras and so on, is that to do their job they're exposing LAN-facing servers, UPnP servers. And it's this server in which a highly prevalent problem has been found. So I don't mean to minimize that. Not at all. I'm hoping that they're not also publicly exposed. None should be.

But so it's important that this is all LAN-facing. In other words, an attacker needs to already have some foothold inside the network to be able to abuse this internal LAN side, LAN-facing aspect of UPnP. I think it is widespread, but it's an internal issue. And so that's why, in all this coverage, we do see references to DLP, Data Loss Prevention, because now one of the things that enterprises are doing is they have this DLP, Data Loss Prevention technology which is explicitly there to catch inadvertent exfiltration of the corporate golden goodies. What happens is this problem, this SUBSCRIBE flaw, allows these UPnP devices, and there may be many in a corporate environment, to be turned into a proxy and used to route an attacker's exfiltration traffic out from the

Intranet onto the Internet. So that's bad. But again, it only happens with an attacker who's already behind the firewall.

On the other hand, remember those problems that we ran across where people on the Internet, external, outside, were able to print something on a printer? Well, that meant that from the outside they were able to get onto a printer, and a printer is on the LAN. And a printer is almost certainly a Universal Plug and Play hosting device. So there's that. So the related question is what about the WAN-facing Interface of UPnP devices? That's what I enhanced ShieldsUP! back then to detect. But that was only one specific vulnerable aspect of Universal Plug and Play. What no one has made clear is whether this SUBSCRIBE vulnerability exists on the WAN-facing side of UPnP devices. And if it does, what would that allow attackers to do?

In the best case, attacks would be limited to using the exposed UPnP device for various forms of reflection attacks. In the worst case, it would allow a remote attacker to probe through the device and into the LAN behind it. But at this point there's no clarity about that. Among the potential problems created by this, the researcher does state, he says: "Scanning internal ports from Internet-facing UPnP devices." So that's certainly not good. And Shodan is still showing lots of Internet-facing UPnP devices.

And as I mentioned, ShieldsUP!, if you are a user of ShieldsUP!, you can take the instant UPnP test. And it won't tell you about this, but it will tell you about the problem before. And if I weren't really busy with SpinRite, I would enhance the test. But I'm sure that will happen without me, elsewhere, soon. I don't even know that this is a big problem. But the LAN vulnerability side is clearly a problem for IT departments. End-users, maybe not so much. Again, something bad has to already be in, in order for that to take advantage of the LAN-facing side.

But to help with determining that, the researcher who found the problem has posted a Python-based vulnerability scanner on GitHub. I've got the link in the show notes. He explains that the script performs a series of actions: finds all UPnP devices on the LAN; finds all UPnP services being offered by those devices; finds all subscription endpoints on those services; then sends these endpoints, encrypted, to a verification server via the UPnP Callback. And then he says: "Servers can't see the endpoints because all encryption is done on the client side. Then this gets the encrypted service list from the verification server and decrypts on the client side, compares the found UPnP services with verified ones."

So to me this is a little bit unclear. But I was a bit disturbed by the line "Sends these endpoints, encrypted, to a verification server via UPnP Callback," and "Server can't see the endpoints because all encryption is done on client side." That may be true, but assuming that the server is on the public Internet, it certainly does see the public IP that's sending it those queries. Since this was somewhat disturbing, I decided to fire the script up on my own network this morning to see what was going on.

Upon running it, I received, and I have it in the show notes, first it posts Stranger Host: http://20.42.105.45, which is certainly not my IP, or any of mine. So that is an IP out on the public Internet. Stranger Port: 80. So he set up an HTTP web server, apparently to field the encrypted results of this client, this Python client. Then he says: "No UPnP device found. Possible reasons: You just connected to network." No. "The UPnP stack is too slow. Restart the script." Probably not. "UPnP is disabled on OS." Very likely. "UPnP is disabled on devices." Very likely. "There is no UPnP supported device." Okay. "Your OS works on VM with NAT configuration." No. My OS is directly connected.

So as I said, the Stranger Host line does indeed appear to indicate that anyone running this script will be sending stuff to that public IP. But even so, my results appeared to all be negative. Since I was assembling this podcast, I didn't want to spend too much time digging into this, but I was skeptical that my network had zero UPnP devices. So I enabled the UPnP service on my FreeBSD-based pfSense firewall. And I ran a different UPnP scanner. It found a UPnP service there on pfSense and provided a truly disturbing amount of information about what it could see. But again, it's on the internal side.

Among that information was the happy news that pfSense does use the miniupnp implementation that, as I said earlier, is known to be unaffected by this flaw. But as I noted, I always had that UPnP server turned off anyway. I then reran the Python script, and it still found nothing. So I'm kind of unimpressed so far. But given the truly horrifying example this guy has posted on GitHub of what his script might put out for somebody in a corporate environment, it might still be worth running. Or maybe it'll turn out that it's possible to do all this locally, as I strongly suspect is probably the case.

So maybe we should just wait for an update of the script that sets up a local server and does it locally because people are not going to like sending this off to some random IP on the Internet, basically a vulnerability report of their internal network. Maybe not. He says it's all encrypted. Okay. I don't know what that means. But anyway, it's in Python. So the source is there. I just, you know, I did this this morning, and I didn't want to take too much time on it.

So where does this leave us? It leaves us where we always find ourselves after something like this. Our networks have connected devices, many of which will likely never be fixed. None of the known problems are critical enough to force us to disconnect them from the 'Net. But we're left with a mild discomfort that things are not as secure as we would like them to be. The lesson this continues to reinforce is that anything that connects needs to have a means for being updated as problems in it are discovered. Everything should have some sort of home, and everything should periodically phone that home to connect for any important updates. In other words, updating is every bit as important as connecting. One should not happen without the other. We're not there today, but that's where we need to aim.

So this sounds like a protocol-level problem. The protocol is being updated. The common wisdom here would be to look for updates to your router firmware. And Microsoft, apparently this is a problem in Windows 10. So they'll be fixing that. And, I mean, every other router on Earth apparently is currently affected. pfSense isn't. That's nice. Anything that is miniupnp-based isn't. I noted that DD-WRT isn't there. Maybe they're also using miniupnp. I don't know. But I'm sure more information will be emerging over time. I'll keep our readers informed.

So Microsoft has started to replace the old Edge with the new Edge. We've been waiting for that to be happening for quite a while. I have Edge on my Win7 machine, and I do think it's a nice browser.

Leo: You like it?

Steve: Yeah. Yeah. I mean, Chromium. It's a nice implementation. And they really - Microsoft really does seem to be focused on adding lots of features. I've got it because I can't wait for the vertical tabs to happen, in which case I will play with those. So everyone using Windows 10 1803 or later will be seeing this appear. It's being rolled out under their Knowledge Base number 4559309. And it will replace the original Edge browser on Win10 2004, 1909, 1903, 1809, 1803. As I was looking at this I was thinking, and isn't it unifying that we only have one Windows 10 now. It's like, uh-huh. Right.

So interestingly, whereas the original Edge could be removed, Microsoft says: "The new Microsoft Edge does not support removal." And they seem to be getting a little more strict about this. Also: "The current version of Microsoft Edge," they said, "will be hidden from UX surfaces in the OS." The current version. So for some reason they don't want to be showing - maybe they're worried that the numbers are getting too big, you know, because they're tracking Chromium's versioning, and it's like, you know, 84, 85. It's like, okay.

Anyway, they said: "This includes settings, applications, and any file or protocol support dialog boxes; and attempts to start the current version of Microsoft Edge will redirect to the new Microsoft Edge." So they're just really pushing it aside, the old one. Of course all previous user data from earlier Microsoft Edge versions - passwords, bookmarks, open tabs, et cetera - will be moved over into the new Edge. So people may notice, oh, look, it sort of feels a little more crisp than the old one did.

But anyway, also separately, there was also some reporting that I saw, but it didn't rise to the level of a bullet point here, that the Win10 Start Menu had begun advertising the new Edge whenever anyone appeared to be searching for information about any other web browser. I very much like the idea of a Chromium-based Edge, and I suppose it's Microsoft's right to push whatever they wish to since they own the platform. But it certainly doesn't give me or any who are reporting on this the warm fuzzies.

But speaking of warm fuzzies, we have some listener feedback. Luis Cruz tweeted: "Barring the ability to transfer your consciousness to a new vessel, do you have a plan for SpinRite and your other work after you are incapable of maintaining it? Will it go open source? Are you mentoring Gibson 2.0 behind the scenes to pick it up?"

Leo: That's a good question.

Steve: It is a good question. And no, I will give it to the world. I will - I don't know what I'll do. Create a GitHub account. Actually, I have one. I just never put anything there.

Leo: Of course, by then there'll be no one who knows how to write in assembly code, so...

Steve: That's true. Well, it'll be a curio. It'll be like the Apollo 11 code, Leo.

Leo: Yeah, exactly. Look how much he did with so few, so few lines.

Steve: Oh, look at that. Five instructions and it talks, yeah. So anyway, I do. I have thought about it. Once I am no longer in a mode where I'm maintaining the code, it's like, why not? I mean, some of it embarrasses me. Some of it is like lots of evolution and stuff. But, you know, when you get older, Leo, I think that's one of the things that happens. You know, Gramps is kind of hard to embarrass now.

Leo: Yeah.

Steve: So that'll be okay. Skynet, he's @fairlane32 is his Twitter handle, he says: "Hi, Steve. In Episode 769" - so that was last week - "with automatic downloads in Chrome, you have the option of choosing 'Ask every time' under Chrome's settings. Wouldn't that prevent the drive-by downloads on sites?" And I said, yes, good point. But it turns out it's not the default, and it's a bit buried. You've got to go to Privacy & Security, and it's not there. Then you've got to go to Advanced because, yes, turning off drive-by downloads is advanced security. But there's a toggle there. I did it, and sure enough, my Chrome is no longer surreptitiously downloading things. It now prompts me. It's a little jarring, actually. I was like, what? Oh, yeah. Well, I did tell it I wanted it to ask me. So I'm happy for that. And you're right, Skynet, it is there.

Chris Rhodus tweeted: "Hi, Steve. Over the years I've heard you say that there is no reason to limit the maximum size of a password." Yes, I was just ranting about it last week. "I'm currently reviewing a vendor design document that has the password max limit set to 32 characters. Are there any case studies or other documents you can point me to that can be used to justify the removal of the 32-character limit? I don't think the vendor will be happy about removing the max limit. I will need to justify the removal of the imposed limit if any opposition is encountered. Thanks."

So Chris, no. I know of no studies. And frankly, 32 characters, you know, unless they're all lowercase "a," it's probably pretty good. And in fact, even if they are all lowercase "a," well, no, I guess somebody could probe to see that there was a 32-character limit, and then try 32 a's.

Leo: That'd be the first thing you'd try.

Steve: Yeah. But the point is - and that's a good point. If there is a limit, then it is a probeable limit. And that does help somebody doing brute-force guessing.

Leo: Right. That's a little weak, yeah.

Steve: So there's something.

Leo: But any rules, any password rules help in that regard.

Steve: Yes. That is exactly right. Maybe that's, well, I was going to say maybe that's why you don't complain until they submit one that breaks a rule. Then you tell them the rule. But on the other hand, that allows it to be probed.

Leo: Right, right.

Steve: So I don't know. I mean, this is a mess. Part of my SQRL spiel is that usernames and passwords is the way I logged into a Hazeltine terminal when I was at Berkeley in 1973, and we're still doing it now. What is wrong? But, yeah. So Chris, I wouldn't upset them. You want them to like you. There's nothing wrong with 32 characters. That's a lot of entropy. If you actually use 32 characters of entropy, that's more than the 256-bit hash that you will reduce that to; right? So that allows as much entropy as, well, actually it's the same; isn't it. 32 bytes is 256 bits. So it's the same, it is the same amount of entropy as if you were to use all bytes in 32 bits. Is it 32? Yeah, it is. So, yeah, that's a lot of entropy. It's enough. So anyway...

Leo: So ones where it's limited to eight characters, you've got to really start to worry about them.

Steve: Yeah.

Leo: And there are those.

Steve: There are. Twelve, you know, not good. There you suspect they're still running - maybe they purchased the mainframe from CompuServe. And, you know, that's not good.

Ed McKiver says: "Steve, just passing a note to add to other comments you might be getting. My Dell laptop auto-updated to Win10 release 2004. Ever since, my computer has been running EXTREMELY SLOW [all caps], and sometimes clicking on Windows items or right-clicking menu takes minutes to come up. Went into Settings to check for updates, and the Settings window crashed during the check before finishing. Second attempt completed okay, but other items like View Update History is taking forever to display. Click and wait and wait and wait is the order of the day now."

He says: "I have 150GB of free space on my hard drive. My laptop tends to slow down when I get under 100GB left." Okay. Now, just let's stop there. What is wrong with this picture? Anyway: "But it's always worked normally with at least 100GB of free space." Good to know. "Thought I'd put in my two cents for stopping all this major release stuff to Windows, if they are constantly going to be breaking stuff."

He says: "Oh. Settings window just crashed again as it was just trying to View Update History." He says: "I can't even roll back at this point. FYI, Ed in Redlands, and SpinRite owner since Version 1." So Ed, thank you. And I think you can probably do - can't you intercept the boot with F6 and get it there, maybe, to roll you back? I don't know. What a mess.

Speaking of a mess, Leo, I have two grc.sc shortcuts to share. These will take users to endcoronavirus.org/states and /countries, respectively. And in case people forget that, grc.sc/states, grc.sc/countries. I really like these two pages because they are all in one place thumbnails. In fact, what you put on the screen right now is really kind of cool. It's a map of the U.S. where the shape of each state is filled with a tiny chart of known coronavirus infections for that state. And they're red when they are not doing so well; green when they've got it under control.

But you can also scroll down and get a divided-by - they show green, yellow, and red with larger historical graphs. And it's just cool because it's a very quick, at-a-glance style, you know, where do we stand by state, and then the grc.sc/countries is where do we stand by relative countries. And in fact you can see down - there you are in the red now. There are some real problems. Arizona has had a really precipitous spike. So has Alabama. It's necessary also to take a look at their absolute number because these are all scaled relative to themselves and not based on the absolute number. So in some cases, you know, when you get some crazy lift-off, yeah, it went from one to five, so whoa, five times. But, yeah. Not that big a problem in absolute terms. But it is just sort of a neat site. I just wanted to share it with our listeners.

I don't have a lot of exciting news to share on the SpinRite project. Probably I'll have more next week. We found a few remaining problems with the FAT partitions I was creating, in the size of their root directories. I had to tweak that since my own, you know, I wrote my own partition formatter from scratch so that I would have ultimate flexibility for the future. And there was a little debris to shake loose. We have some people who have created batch files to create uniquely named files until it crashes in order to torture test these partitions that I've created, and they are now passing with flying colors. So I thank those listeners in our newsgroup. And that's all fixed and has been torture tested.

It was suggested that we needed to further failsafe or add a failsafe means for selecting the drive whose contents we are about to permanently wipe out and destroy through a destructive reformatting. So I added the technology with the utility I'm creating that's called InitDisk. And it watches all of the system's drives, asks the user to confirm the drive they wish to blow away by physically removing and replacing it. And that worked quite well.

But it worked so well that I then wanted to see how it would be if we used that as the only means for specifying the drive we want to reformat and use because there were some problems. If you put a stick in, a USB drive that, for example, had been over on a Mac and only had a GPT format and a partition that wasn't recognized by Windows, it wouldn't assign a drive letter. So you weren't able to use a drive letter to point to the drive you wanted to reformat. Anyway, I'm currently working on using pure physical insertion as the means of specifying the drive. It'll tell you all about the drive, and then you'll confirm that that's the one you want to reformat.

And it also turns out that there are some drives - for a while Microsoft was requiring USB drive vendors to have their drive declare itself fixed, rather than removable, in order to be certified under Windows 8. SanDisk was apparently the vendor that immediately jumped on that. And so there's a period of time when they were making USB drives that, when you stuck them in, they said they were fixed. Well, I had a prevention of only allowing removable drives because I didn't want there to be no possibility that a user could inadvertently reformat one of the fixed drives they had in their computer.

Well, it turns out that wasn't reliably useful because of this weird Windows 8 certification issue that went by. So now we're just going to use, you know, prove to me by inserting it that this is the one you want to use, with then follow-on verification. So that'll be done probably tomorrow, and then we'll be moving forward again. So anyway, working on this project, and we're getting a lot done there.

Okay. Zoom's end-to-end encryption debacle. Yes, just when everything appeared to be going so well, the day after our podcast last week which, as we all know, celebrated the quite rational and well-supervised planned evolution of Zoom's security architecture, during a call with financial analysts to discuss Zoom's latest financial results, CEO Eric Yuan confirmed that Zoom won't be offering end-to-end encryption on free accounts. What?

So Eric was widely reported to have said, and I quote from multiple sources: "Free users, for sure, we don't want to give that [end-to-end encryption] because we also want to work it together with FBI and local law enforcement, in case some people use Zoom for bad purpose." Oh, boy.

So not surprisingly, my Twitter feed lit up with our listeners asking whether I had seen this latest from Zoom, and many saying that no way are they going to be using it from now on. To give everyone a sense for the industry's reaction to this revelation, The Verge headline: "Zoom says free users won't get end-to-end encryption so FBI and police can access calls." The Guardian: "Zoom to exclude free calls from end-to-end encryption to allow FBI cooperation." Engadget: "Zoom explains why free users won't get end-to-end encrypted video calls."

USA Today, so it's even out of the tech press: "Zoom CEO: No end-to-end encryption for free users so company can work with law enforcement." The Next Web: "Zoom won't encrypt free calls because it wants to comply with law enforcement." Tech Crunch: "Zoom faces criticism for denying free users end-to-end encryption." And I could keep going, but you all get the idea.

The problem, of course, is that this is seen as purely a profit motivated policy, since strong end-to-end encryption is desirable, and only paying customers can get it. And the argument about compliance with law enforcement, well, what? So you want to allow Zoom to comply with law enforcement except if people pay for the service. In which case Zoom's newer and better end-to-end encryption will explicitly not allow for any compliance with law enforcement.

Okay. So you're making money by marketing your hostility to law enforcement. Of course we've seen that before. That's the stance that Apple has explicitly and loudly taken. But Apple's encryption is ubiquitous. Apple doesn't allow anyone to not have full end-to-end encryption on any of their person-to-person connections - text, voice, or video.

So of course I was as stunned as anyone by this news, especially given the mature supervision that Zoom was apparently now receiving. But the CEO had spoken, and on this he seemed quite unambiguous. So I thought I'd reach out to Alex Stamos to see what he might know. I brought him up in TweetDeck and discovered that he follows me on Twitter. So that meant I could DM him directly and hopefully not be lost in the noise. I shot Alex a DM to make sure that he was aware of this mess.

And then I continued poking around and quickly discovered that he was quite well aware indeed. He had posted a series of tweets Tuesday, the previous day, that attempted to repair the damage. But frankly, they only further complicated things. I have a link in the show notes to his Twitter stream, which is not too long, so I'm going to share it because it just demonstrates, what?

So here's what he said. Alex Stamos tweeted: "Some facts on Zoom's current plans for E2E encryption, which are complicated by the product requirements for an enterprise conferencing product and some legitimate safety issues." Next tweet: "All users, free and paid, have their meeting content encrypted using a per-meeting AES-256 key. Content is encrypted by the sending client and decrypted by receiving clients or by Zoom's connector servers to bridge into the phone network or other services." Okay, so far that sounds like everything's encrypted. He says "all users, free and paid."

Then he says: "Zoom does not proactively monitor content in meetings and will not in the future. Zoom doesn't record meetings silently. Neither of these will change. Our goal is to offer an end-to-end encrypted solution that provides a stronger guarantee. Zoom is dealing with some safety issues. When people disrupt meetings, sometimes with hate speech, CSAM" - whatever that is - "exposure to children, and other illegal behaviors, that can be reported by the host." Right, because they're in the meeting. "Zoom is working with law enforcement on the worst repeat offenders." Good.

"Making it possible for hosts to report people disrupting their meetings even under end-to-end encryption is solvable." Yeah, we've talked about that. "The likely solution will be a content ring-buffer of the last X seconds on the host's system that can be submitted to Zoom for triage and action." That's entirely reasonable. That's the host does that from within the cone of silence.

"The other safety issue," he tweets, "is related to hosts creating meetings that are meant to facilitate really horrible abuse. These hosts mostly come in from VPNs, using throwaway email addresses, create self-service orgs, and host a handful of meetings before creating a new identity. Zoom's Trust and Safety team can, if they have a strong belief that the meeting is abusive, enter the meeting visibly and report it if necessary." Okay, that's new.

He says: "As you see from the E2E design, there is a big focus on authenticating both the people and the devices involved in end-to-end meetings. If properly implemented, this would prevent Zoom's employees from entering a meeting, even" - okay, wait, now. "As you see from the E2E design," he says, "there's a big focus on authenticating both the people and the devices involved in end-to-end meetings. If properly implemented, this would prevent Zoom's employees from entering a meeting, even visibly. There will not be a backdoor to allow this." Okay. So sounds like we're talking about straddling technology. Currently, or until this is properly implemented, Zoom can do this. They won't be able to in the future with a proper implementation.

He says: "Zoom's E2EE implementation will need to be opt-in for the foreseeable future. A large portion of Zoom's meetings use features that are fundamentally incompatible with end-to-end encryption - telephones, SIP phones, room systems, cloud recordings, cloud transcription, streaming to YouTube, et cetera. So we have to design the system to securely allow hosts to opt into an end-to-end meeting and to carefully communicate the security guarantees to hosts and attendees. We are looking at ways to upgrade to E2E once a meeting has started, but there will be no downgrades.

"So this creates a difficult balancing act for Zoom, which is trying to both improve the privacy guarantees it can provide while reducing the human impact of the abuse of its product. Lots of companies," he says, "are facing this balancing act; but as a paid enterprise product that has to offer end-to-end encryption as an option due to legitimate product needs, Zoom has a slightly different calculus. The current decision by Zoom's management is to offer end-to-end encryption to the business and enterprise tiers, but not to the limited, self-service free tier." He says that's the current decision by Zoom's management is business and enterprise tiers, but not to the limited, self-service free tier.

He says: "A key point: Organizations that are on a business plan, but are not paying due to a Zoom offer, like schools, will also have free end-to-end encryption. Will this eliminate all abuse?" he asks in his tweet? "No, but since the vast majority of harm comes from self-service users with fake identities, this will create friction and reduce harm. This," he tweets, "is a hard balance. Zoom has been actively seeking input from civil liberties groups, academics, child safety advocates, and law enforcement. Zoom hopes to find a common ground between these equities that does the most good for the most people." Good luck with that.

He finishes with three final points: "One, most of the people I interact with know this, but I've been working with Zoom as a consultant and helped with the E2E design. Two, none of the major players offer E2E by default - Google Meet, Microsoft Teams, Cisco Webex, BlueJeans." He says: "Webex has an E2E option for enterprise users only, and it requires you to run the PKI [Public Key Interface] and won't work with outsiders. Any E2E shipping with Zoom will be groundbreaking. Three, at no time does Zoom turn over encryption keys to law enforcement. The issue here is whether Zoom's own employees can enter spaces they host." Wait, the employees host?

Leo: Zoom hosts.

Steve: I don't know who "they" mean.

Leo: But Zoom has the keys, in other words. "They" meaning Zoom.

Steve: Well, elsewhere he also said they don't.

Leo: So how would he hand them to law enforcement if they didn't have them?

Steve: He says: "At no time does Zoom turn over encryption keys to law enforcement." Oh, meaning they have them, but they don't hand them over.

Leo: They have them. They don't hand them over. And that's why employees could listen to any meetings Zoom hosts.

Steve: So he says: "The issue here is whether Zoom's own employees can enter spaces they" - he doesn't say who "they host" is.

Leo: Zoom.

Steve: He says "they host."

Leo: Zoom.

Steve: Is that...

Leo: Yeah.

Steve: So because Zoom's hosting all Zoom meetings.

Leo: Right.

Steve: Okay.

Leo: Employees don't host meetings.

Steve: Right. Well, right. But we had this notion earlier that the host of the meeting could report bad activity. And that's fine.

Leo: Right. So who cares what Zoom employees' meetings can or can't do.

Steve: Right.

Leo: That's not really any issue.

Steve: Okay. So he posted all that on the 2nd. Two days later, on Thursday the 4th, it was contradicted by a Zoom spokesperson who confirmed that free users will be covered by Zoom's AES-256 GCM encryption, but chats will not be covered by additional end-to-end protections. So the official Zoom spokesperson said: "Zoom's AES-256 GCM encryption is turned on for all Zoom users, free and paid. Zoom does not proactively monitor meeting content, and we do not share information with law enforcement except in circumstances like child sex abuse."

Leo: Yeah, they have the key. Right.

Steve: "We do not have backdoors where participants can enter meetings without being visible to others. None of this will change. Zoom's end-to-end encryption plan balances the privacy of its users with the safety of vulnerable groups, including children and potential victims of hate crimes."

Leo: Inherently, they just said two completely incompatible things.

Steve: I know. Exactly.

Leo: Okay.

Steve: "We plan to provide end-to-end encryption to users for whom we can verify identity, thereby limiting harm to these vulnerable groups. Free users sign up with an email address, which does not provide enough information to verify identity. The current decision by..."

Leo: It's not unreasonable, I think.

Steve: "The current decision by Zoom's management is to offer end-to-end encryption to business and enterprise tiers. We are determining the best path forward for providing end-to-end encryption to our Pro users. Zoom has engaged with child safety advocates, civil liberties organizations, encryption experts, and law enforcement to incorporate their feedback into our plan. Finding the perfect balance is challenging. We always strive to do the right thing." So in my show notes here I said, uh, okay. So today we still have no idea what they intend.

Leo: They got the headlines they wanted, though, didn't they.

Steve: They did. They are clearly asked, over and over, do you or do you not encrypt all Zoom video? And they answer, "Right."

Leo: Good one. Yeah, that's right.

Steve: Uh-huh. Uh-huh. You got it.

Leo: Yeah, yeah, [crosstalk].

Steve: Do you or don't you? Right.

Leo: Right.

Steve: If you have been listening carefully, yes. Yes what? Uh-huh. But Leo, I have no idea.

Leo: We don't know what they're doing.

Steve: None whatsoever.

Leo: It's unclear. And Stamos didn't clear it up.

Steve: No. His is as bad as the official spokesperson. And they end with, yeah, this is hard. Yeah. Okay. Sorry.

Leo: I don't - I think, you know, I didn't have the same reaction a lot of people did to this idea of encrypt for paid users, but not for unpaid users, because it's pretty clear an unpaid user can use an account without proof of personality.

Steve: Yes. Yes.

Leo: So it would have been better had they said we'll encrypt, but only for people who prove their identity, give us identity, so that we can chase down malefactors because that's been a problem on our platform. But they're not even saying that. It's completely unclear. Basically they can give law - it sounds like they can give law enforcement anything because they have the keys.

Steve: And they're saying, I mean, Alex said it and the spokesperson said it. Encryption is turned on for free and paid. And then, but not so much.

Leo: And what? Huh?

Steve: So maybe everything's encrypted, but unless you pay they hold the key? I don't...

Leo: It sounds like they have the key to everything. That's what it sounds like.

Steve: Yeah.

Leo: In which case this whole thing was FUD of Eric Yuan saying, "Well, we want to be able to have law enforcement tap the unpaid members." Well, they can tap all of them because, if Zoom has the keys, they just give you a warrant. Maybe they aren't going to ask for a warrant on the - maybe that's it. It's just not clear what they're saying.

Steve: That's a mess. So thank you for clearing that up.

Leo: Yeah. And this actually leads me to kind of - this is more to me of the same obfuscation and hand waving they've been doing for years. And it really - I am liking them less and less. They bought the best names in security, but it doesn't sound like they did much with it.

Steve: No.

Leo: But that's just my personal opinion because we don't know.

Steve: Yeah. I mean, and if the design that we discussed last week is actually implemented, then that would be good. But now it looks like no one is sure. Even Alex said "if it's implemented correctly."

Leo: Right.

Steve: He gave himself kind of, oh, so wait. It might not be?

Leo: He's not doing the implementation.

Steve: No.

Leo: Nobody, none of these big names they hired is implementing. They wrote that nice paper. And then they said, "Here, Zoom. We deliver this to you. You go to your Chinese programmers and make it so." And that's, I mean, we're done. We're done here. I'm sure they all got a ton of money. I honestly - this does not reassure.

Steve: And that's why Alex, you could read him saying, "I hope you understand I'm just a consultant here." You know? Point number one at the end was he said: "Most of the people I interact with know this, but I've been working with Zoom as a consultant and helped with the E2E design." He's like saying, you know...

Leo: I wrote that paper, yeah.

Steve: Yeah.

Leo: You know, I'm also going to downgrade Alex Stamos a little bit here. It was always a little weird that he left Yahoo because, he said, "They never told me about the billion-person breach." Goes to Facebook, "They never told me about the Cambridge Analytica." Goes to Zoom, "Well, it's not my implementation." So it downgrades you, too, a little bit, Alex. I expect a little better than that. Oh, well.

Steve: So I felt it was important for our listeners...

Leo: I'm glad, yeah.

Steve: ...to join me in FUD because, like, I don't know what, I mean, nothing could be less clear. This, you know, I have no idea what they're doing now.

Leo: I think it's intentionally obfuscated. And don't forget the most important part of that statement, "Well, no one else does it, either."

Steve: Right.

Leo: Right? Well, what do you want? No one else is doing it for free.

Steve: And if we did, it would be groundbreaking. But, you know, maybe we won't. Oh, okay.

Leo: We have to do it in accordance with law enforcement and good practice. They don't want to get in trouble with Bill Barr. Honestly, how important is it that you have end-to-end encrypted video conferences? If you're doing - I guess if you're doing, you know, you're a psychiatrist having sessions with a client, that would be pretty important.

Steve: It's got to be HIPAA-compliant in order to do that.

Leo: Right. Right.

Steve: And, now, here's the problem. If you're an enterprise user, and you're doing transnational video conferencing, discussing trade secrets...

Leo: There you go.

Steve: ...are you going to trust Zoom? I'm not going to trust Zoom now. No. We're going to use some sort of - Lord knows what. How do you do truly - you use FaceTime.

Leo: It's hard to do, isn't it. It's hard to do. Point-to-point's easier than distributed.

Steve: Yeah.

Leo: I'm so glad you brought this up. And you've actually clarified my thinking because I didn't see this actual statement from them, which is bonkers.

Steve: I know.

Leo: It's bonkers. Okay. Steve Gibson. See, this is why we've really got - we've got to give him a medal. But, yeah, some sort of a major award because, without him, I don't, you know, this stuff would go right by. You certainly don't see any discussion of it in the mainstream media. GRC.com, that's where Steve hangs his hat. The Gibson Research Corporation's where SpinRite, the world's best hard drive recovery and maintenance utility is currently under construction, the new version. Old version available. Buy it now, you'll get in on the beta, the next version.

He also, of course, puts the podcast there. He does a couple of things no one else does. I don't. 16Kb versions, which sound a little like Thomas Edison on his phono disks. But they're very small. And that's their benefit for the bandwidth impaired. There's also transcripts, the smallest yet. And he pays good money to Elaine Farris to listen to our blather and write it all down perfectly. So those are really useful. They're also searchable, which makes it even more - that's really the best reason for it.

He also has 64Kb audio. That's all at GRC.com, along with all sorts of other freebies like SpinRite. No, that's not free. ShieldsUP!, that is, now with built-in testing for Universal Plug and Play flaws. You can also get all sorts of other stuff there. It's fun. It's a rabbit hole you go into and you never come out. All sorts of fun things to read.

You can also catch this show on demand on our website, TWiT.tv/sn. It's on YouTube. You can ask your voice assistant to play the Security Now! podcast. They'll play the most recent episode. If you want to watch live, we do it Tuesdays round about 1:30 in the afternoon. That's 1:30 Pacific, 4:30 Eastern, 20:30 UTC. The live streams - and they were little glitchy today, I apologize. People asked us to rewind the show, but we can't do that. The live shows are at TWiT.tv/live. But the beauty part is we've got it on demand, so you can see the whole thing you missed in just a matter of a few hours. If you subscribe, you'll get it automatically. That's probably the best thing to do. Find your favorite podcast application.

Steve, stay well, stay healthy, and stay on top of things for us because we'll need you next week on Security Now!.

Steve: I'm ready. See you next week, buddy. Bye.