Windows 7 - R.I.P.

Description: This week's Security Now! podcast is titled "Windows 7 - R.I.P.," not because there's much that we haven't already said about the fact, but that it happens TODAY; and that, given the still massive install base of Windows 7, it's significant that all of those machines will now be going without any clearly needed security updates. So the big news for this week WAS to be the event of the first successful preimage attack on the SHA-1 hash. But that news was preempted at the last minute by the much more immediately significant news of the remotely exploitable "Cable Haunt" vulnerability that's present in most of the world's cable modems right now! So we'll be talking about that after we look at the FBI's recent request to have Apple unlock another terrorist's iPhone; update on the Checkrain jailbreak solution; examine the challenge of checking for illegal images while preserving privacy; look at some deeply worrying research into just how easy it is for bad guys to get SIMs swapped; examine the consequences of not patching a bad VPN flaw; deal with a bit of miscellany; and then, finally, look at the new "Cable Haunt" vulnerability.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-749.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-749-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with a fond farewell to Windows 7. Today's the last update. And, man, it's a good thing, too, because there's some major issues with all versions of Windows that Microsoft is fixing today. And then he's going to terrify us with Cable Haunt, an exploit that is going to affect cable modems all over the world and is very difficult to fix. In fact, you can't fix it. Only your Internet service provider can. Details coming up next on Security Now!.

Leo Laporte: This is Security Now!, Episode 749, recorded Tuesday, January 14th, 2020: Windows 7 - R.I.P.

It's time for Security Now!, the show where we cover your privacy, security online with this guy right here, Steve Gibson. Hi, Steve. Haven't been here since the New Year, but it's nice to see you.

Steve Gibson: Leo, likewise. Great to see you, always.

Leo: Did you have a good New Year?

Steve: Yeah. Just a quiet time.

Leo: No dancing with Captain Kirk or anything?

Steve: My sort of favorite kind of New Year. Well, I must say I do miss the old days...

Leo: I do, too.

Steve: ...of TWiT craziness and butt tattoos, and you just never know what's going to happen.

Leo: You know what I found out? Those tattoos never go away. My hair grew back, but the - still there.

Steve: Yeah. The good news is this came as no surprise to your wife. So it's not like one day you were walking away in the buff, and she said, what? What is that little...

Leo: Yeah, she was present at the inception, as we say.

Steve: Yeah. Is that an AND gate with a personality, or what is that?

Leo: Instead of doing that, we went to CES, had a lot of fun at the Consumer Electronics Show, formerly known as the Consumer Electronics Show. And I want to thank Jason Howell for filling in last week. But I don't plan to go anywhere for at least six months. I'm here.

Steve: Whoa, what? Oh, come on. Not till summertime?

Leo: That SQRL logo would make an awfully nice tattoo. I'm just looking at...

Steve: Okay, no.

Leo: It's kind of made to put on your butt.

Steve: So speaking of butts, this podcast is titled "Windows 7 - R.I.P."

Leo: Oh, boy.

Steve: Not because there's much that we haven't already said about the fact. We've pretty much beaten that topic to death. But just because that happens exactly today, and I just couldn't pass up the opportunity to say, well, I mean, it is significant because there's still a massive install base of Windows 7. We're going to talk about it just briefly at the end because it turns out it's a little more than one in four workstations, the most recent survey is, are still running Windows 7. Despite the fact that you could argue

Microsoft hasn't quite got it fixed yet, though they've been trying since, what, since '08 was it, I think, that it came out?

Leo: This is Patch Tuesday, so today's the last Patch Tuesday for Windows 7, yeah.

Steve: Yes. Yes, it is.

Leo: And just in the nick of time, too, by the way.

Steve: Well, I'll talk a little bit about my plans because over the last five years I've warmed up to Windows 10. I don't like it nearly as much as 7, but I don't hate it. I've sort of - I've capitulated like the rest of the world has. So that's the title of the show. The big news for this week was to be the event of the first successful chosen prefix collision on the SHA-1 hash, which really marks its demise.

But big as that news was, it was preempted by the last-minute revelation of a remotely exploitable, kind of, but I'll explain, exploit against probably all cable modems. And there are a lot of those. The team that did this is based in Europe, and they found 200 million cable modems that were exploitable. Probably all of those in the U.S. are, too. So they named this "Cable Haunt." And as one does these days, they grabbed the domain name, and someone came up with a good logo for it. And because it's an issue now, and there is something that - well, I learned a lot, actually. I found out that I have a browser-accessible spectrum analyzer in my cable modem. And so do you, Leo.

Leo: What?

Steve: Didn't know. It's really cool. You can dynamically see the spectrum of data coming in through your cable modem.

Leo: Wow.

Steve: So it's like, what? Anyway, and it turns out, though, that's kind of a mixed blessing because there's a problem with it.

Leo: No.

Steve: Which could be exploited. But anyway, we're going to talk about that. We're going to take a look - and I just heard you mentioning it toward the end of MacBreak Weekly, talking about the FBI's recent request to have Apple unlock another terrorist's iPhone and what the downstream consequences may be of that. We're going to check in on the Checkrain jailbreak solutions, speaking of iPhones, and see how that's coming along; examine the challenge of checking for illegal images while preserving privacy; look at some deeply worrying research into just how easy it is for bad guys to get their SIMs swapped, that is, it's shockingly easy. So found a group of four researchers from Princeton.

We're going to examine the consequences of not patching a really bad VPN flaw for the last nine months, and then deal with a little bit of miscellany. I made the mistake, Leo, of going to a 4DX movie. I've never walked out of a Star Wars movie in my life until I did on Friday. So I thought it'd be fun to chat with you a little bit about that horrifying...

Leo: Wait, is there a new Star Wars movie I didn't know about?

Steve: And then, so we're going to deal with some miscellany, and then finally wrap up by explaining what Cable Haunt is and why it actually could be a problem. So, wow.

Leo: Did you see the stuff about the Crypt32.dll problem with Windows, too?

Steve: No.

Leo: I'll give you a chance to look at that because it apparently affects all versions of Windows. Brian Krebs...

Steve: And that's what you meant when you said "just in the nick of time."

Leo: Yeah.

Steve: Do we know if it's being fixed today? Or is this going to be a, oh, gee, don't you really wish you had Windows 10 because then you would get the fix.

Leo: Microsoft has quietly shipped a patch to branches of the U.S. military and other high-value customers/targets that manage key Internet infrastructure. But Patch Tuesday apparently is the day that we're going to see a fix for Crypt32.dll.

Steve: Okay, yup.

Leo: That's the cryptographic function in Windows that is apparently broken and a serious security vulnerability.

Steve: Oh, just imagine that. Thank goodness that now, after today, all of the problems in Windows 7 have been resolved.

Leo: That's it. No more.

Steve: And there just is no more.

Leo: No more.

Steve: We're not getting patches because they finally got it all fixed.

Leo: They fixed it.

Steve: Yeah, right up at the finish line.

Leo: The biggest issue Krebs talks about is it could be used to spoof certificates and digital signatures on software. So malware could pretend to be something legit with a spoofed certificate because of this flaw. So it might be a good day to apply your patches. I'm going to do that right now, actually. Okay, Steve. Let's haunt my cable. Here it is.

Steve: So the FBI has, well, yeah, we do have our Picture of the Week, which is the fun logo that the Cable Haunt guys created. And we will be getting to that at the end because it's horrifying.

Leo: Yeah, no kidding.

Steve: The FBI has asked Apple to unlock another iPhone.

Leo: Oh, yeah.

Steve: And so who knows what this is going to mean. What did you guys conclude over on MacBreak Weekly?

Leo: Well, Apple was incensed because Attorney General Barr had the temerity to say, oh, Apple's not helping us at all, and they listed all the things they've done to help the FBI, including the iCloud account, account information and so forth. But they said, and I think quite rightly so, at the last paragraph of their statement, it's a bad idea to have a backdoor in any encryption. It's just going to give it to the bad guys, as well as the FBI; and we just don't think it's a good idea. We don't want to do it. So we have - they were at great pains to say, no, no, we're helping the FBI. We're helping them in every way we can. But we can't give them that encrypted information. We have no way of breaking in, and we don't think it's a good idea to put something in there that can't.

Steve: And so for what it's worth, I wanted to sort of - I know I've confused some people when I've talked about how it's obviously possible for Apple to add a silent listener to an iMessage group. That's very different from the FBI coming to them and saying we want you to decrypt this phone. In this current case, this is the guy who shot up the military base, what, six weeks ago down in Florida, who actually - he had two iPhones, and he shot one of them. One of them has a bullet through it. So that may be a little more tricky to recover, if Apple even could.

But the point is, since the Syed Farook event that we had in 2016 - I guess actually the shooting was in 2015, and then the whole issue happened, kind of boiled over in 2016. Since then Apple has further strengthened the security of the iPhone system. And I am

absolutely sure that they're telling us the truth, and the FBI, when they say we designed a system that we cannot get into. So, I mean, there is the iCloud backup stuff, which sort of creates a little bit of softness to the whole thing. But in terms of the phone itself, I'm sure that there is absolutely nothing Apple can do.

So sort of like the question, the other issue that we have of whether an individual can be compelled to disclose something that they know, like a password, in order to unlock their phone, is that testimonial or not? And we know courts have been going back and forth on this. And ultimately that's the kind of question that may end up getting pushed up to the highest court that we have in the U.S. for a decision. But this issue, if the FBI and Apple are going to face off, this is not something that the Supreme Court can rule on because it may very well be that Apple has, in fact I'm sure it is, they've produced a system they cannot decrypt.

And so what that says is that what would be required is that the U.S. law would end up being pushed, being changed to make it unlawful for commercial encryption systems to be sold in the U.S. that don't have some means for law enforcement to get in. And on that count, I am 100% aligned with all of the crypto experts who say there's no way to do that without weakening the system. And so, again, I wanted to make the point that that's very different from an Apple negotiated real-time, essentially a wiretap on iMessage, which they could do if they wanted to. In this situation I'm sure we've got phones that, notwithstanding the various other corner cases that we've talked about, that Apple cannot get into.

And it'll be really interesting to see if such legislation happens. I'm skeptical that that is going to happen because, I mean, the fact is it really does weaken crypto in a way that I don't think it's clear you can fix. And Apple is sort of a special case. It's also probably not a broad enough perspective to keep talking about them because they really have an iron grip on our phones. I mean, we're sort of renting them from Apple because Apple absolutely decides what the devices will do and won't do, which of course is why some people like to jailbreak their phones in order to get back some of the control that Apple has designed out of the system that we're using sort of with their permission because, I mean, they really are tightly tethered to Apple.

So anyway, I just sort of wanted to touch on that. It'll be interesting to see where this goes. I mean, as far as I know, this issue is pending. The FBI general counsel wrote a letter to the Apple general counsel saying we need help decrypting these phones. The general counsel replied we'd like to help you help you, we really would. We help you in all these other kinds of ways. But the phones are designed with our customers' privacy paramount, and we're allowed to do that, so we have. And the question will be, is that ever going to change, that commercial companies would not be allowed to do that? And we know the arguments are there's lots of noncommercial solutions for encrypting stuff, if that happens. And that would be, you know, a good point to be raised. So, boy, it'll be interesting to see whether this ends up getting legislated in 2020.

Leo: The only way I could see that this would happen is if they decided to criminalize encryption to the point that, if you had it, if you used it, you would go to jail. Because you're right. They can't stop somebody from using Signal or some other open source solution. The math is out there. But they could say it's a violation, it's against the law to use something that has crypto. If we can't see it, then you're breaking the law. And that...

Steve: So they could say that it is against the law for a U.S. citizen to withhold information that would allow law enforcement to decrypt their phone.

Leo: No, no, no. I'm saying they could say to you, Steve Gibson, you may not use encryption; that that in and of itself is a crime. Criminalize encryption.

Steve: Yes. I knew that's what you meant. But that seemed like way far out there.

Leo: It's the only way you could do it because, even if they said to Apple, Samsung, everybody, okay, you know, and okay, my phone is no longer encrypted, I could still use Signal or some other...

Steve: Right.

Leo: Or write my own. I could still use crypto because crypto's well understood. It's out there.

Steve: Right. And so that's why I had backed away from that to the position of it's against the law for a citizen to refuse to provide the information they have to law enforcement to see into their crypto.

Leo: Including your own. That's right, yes.

Steve: Exactly. And of course the one glitch there is, in this particular case, we have a dead terrorist who is unable to divulge the information. And of course users could still choose to go to prison rather than divulge what is in their mind, their ability to decrypt their device, if what's in there is really so bad that prison is the better outcome. I mean, we're really in a mess. This is really a mess. And it will only be if somehow the decision is finally made that companies have the right to encrypt in a way that the government cannot see. And, boy, that's difficult to square. I mean, it's just hard to imagine a world where, not just the U.S., but the U.K., that's even more aggressive about seeing into their citizens' data, yeah, it's just really, for me, fascinating.

Leo: There is some evidence, and I'll have to read more on this, that the cryptographic vulnerability we were talking about in Windows and Crypt32.dll is because it's straightforward to create a second private key without the knowledge of the original signing private key. And you have to wonder if Microsoft might have thought that would be a useful thing to put in their crypto library, a backdoor.

Steve: Huh.

Leo: I mean, I have to read a lot more with it. But this is an interesting question. There may be more to this.

Steve: So it looks like it may not be a bug, it may be a feature.

Leo: Well, we saw this before with the vulnerability with WMF.

Steve: That's exactly where I was going to go, Leo, was I got really taken to the cleaners by people who said, oh, Gibson, you don't know what you're talking about. And I said, look, I'm looking at the code. I can imagine a Windows - this was when Windows was the only thing that ran Windows metafiles. And I could totally see somebody saying, well, we're going to have all this interpretive stuff. But wouldn't it be cool if you could execute code in a Windows metafile? And it was like, back then, before networking and before the Internet and before all this concern, it would have been a reasonable thing to do. And the problem is it stayed in there for decades until someone stumbled on it and said, uh, this looks really bad. And to me it looked like an understandable cool backdoor to allow code to be executed in a Windows metafile. So, yes, Leo, you just hit on exactly the analogy I was thinking of.

Leo: Yeah, well, we'll learn more about it. This is just breaking now. But the NSA has just put out a warning saying patch it.

Steve: Well, and doesn't this make you wonder, I mean, if this isn't - do we know that it's patched today? Is it in all of the Windows OS Patch Tuesdays today?

Leo: It's my understanding that it is, but I haven't seen anything from Microsoft.

Steve: Because this is another example, I mean, we saw XP updated recently when this horrific problem in RDP was found, in Remote Desktop Protocol. So one really wonders how well Microsoft is going to be able to adhere to their refusal to update Windows 7. I mean, imagine that this hadn't happened until after today? Are we saying that Microsoft is going to allow a huge security vulnerability in all Windows 7s? And people who are paying for them still get them. But those who aren't paying, sorry, you have no recourse.

Leo: I just think also there's a certain irony in the fact that it's the NSA warning us about this cryptographic vulnerability when at the same time other members of the administration are saying we want more cryptographic vulnerabilities.

Steve: Yes, good point.

Leo: And the NSA has always had that kind of duality where they're trying to protect us at the same times they're trying to snoop at us.

Steve: I know. There was that dual random number, DRBG, which, like...

Leo: Right. Didn't they get the RSA to weaken encryption?

Steve: Yes. I mean, it really looks like, you know, there were three different random bit sequence generators, and they paid RSA to choose arguably the worst of the three, which came from an unclear...

Leo: So we can crack it.

Steve: Yeah, which sort of just seemed really, again, you don't know for sure. But if it walks like a duck and quacks like a duck, then maybe it is a duck. Anyway.

Leo: I don't want to go all conspiracy theory, but it's fascinating. It's just...

Steve: So Checkrain is the unfixable, actually in the boot ROM. And, boy, I was thinking of that also today because the thing we'll be talking about with our cable modems can be fixed with a firmware upgrade, but end users are unable to upgrade their own cable modem firmware. That has to be something that's pushed from your provider. So but my point is thank goodness that's not in ROM, or it would really be a problem. And it's a big problem as it is. But still, in the case of Checkrain I wanted just to kind of check back in.

It looks like they've now solidly got it working across all iPhones. It's still in beta at 0.9.1. And the most recent beta, bringing us to 0.9.1, fixes multiple bugs, including, they said, an issue where the loader app would crash when installing Cydia on iPads; a crash when the macOS language was set to anything other than English; an issue where iPad Minis would not work with the GUI; an issue with the scp - that's secure file copy - binary not working as expected. And they ended up saying, under "unsupported devices," all there are now are some iPads - iPad Air 2, iPad 5th Gen, iPad Pro 1st Gen. Oh, there is an iPhone 5s, so there's one iPhone that they still - they say it can work, but it may require more attempts than usual. And then the iPad Mini 2 and Mini 3 and the Air.

So basically they've got it. They've got the persistent jailbreak, persistent as in Apple cannot fix it. But remember that it is a boot time thing. So rebooting your device will flush it out of RAM. Still, they're continuing to make inroads, and it is now available on macOS and Windows. And they're still working to bring it up under Linux. So that little bit of soreness continues marching forward.

While we're on the subject of Apple and the privacy of their users, I thought it was interesting that during last week's CES, the Consumer Electronics Show, their Chief Privacy Officer Jane Horvath confirmed that Apple is automatically and continuously scanning images being backed up to iCloud in order to ferret out any instances of child abuse. She explained that this was the way they were working to help fight child exploitation, as opposed to breaking encryption.

And it is the case that there are some technologies, Microsoft has something called PhotoDNA, which is sort of a soft hashing process which makes this sort of thing possible. We know, for example, that hashing is useful for determining the exact duplication of something without revealing what it is. And, for example, everybody, all responsible websites, are now using that for the password-based login. When the user first logs in, creates their account, establishes a password or changes the password, their password is hashed very strongly, and the strength keeps going up, the number of iterations. And there are of course memory-hard ways of hashing that makes it much more difficult to accelerate hashing.

But the point is that password is turned into something that is derived from the password, and that's what's stored. So that in order to check when the person relogs in, the same thing is done, and the results are checked. So you can imagine this being done with images, where you would hash an image, and then you could check for a duplicate image if the hashes of the images matched. And it turns out that for the past 12 years the National Center for Missing and Exploited Children, the NCMEC, has been making available a large file of hashed values for known child sexual abuse images.

So one thing that lots of responsible large companies are doing is, when images are uploaded, they hash them and then check them against this hash list of known abusive

images to see if they get an exact match. Of course the problem is, as we know about hashes, even if one pixel of an image were one shade darker, it would result in a hash having about half of its bits on average inverted. It's like, you know, one bit changes, and half the bits in the hash have a 50% probability of changing. That's what's cool about hashes. So you can't hash images in a useful fashion.

What Microsoft came up with, with the so-called "PhotoDNA," is that the images are resized to a standard size. Then color is pushed out, so they're turned into black and white. Then it's broken into a grid, and some technology scans each region of pixels within each grid cell and generates - uses an algorithm to generate a fingerprint for the edges that are seen within that cell. That ends up producing something which is intended to survive cropping and resizing and recompressing and so forth, so that basically it produces a - it's still using a hash-like approach. Nothing of the actual content of the image survives.

But it does produce a fingerprint such that sufficiently similar images should produce a collision and raise a red flag, and then probably that pops them to someone's attention who would then verify whether or not it looks like it's exploitive. And that's technology which lots of large companies - Microsoft, Google, Verizon, Twitter, Facebook, Yahoo, and so forth - are using in order to basically make sure that they're not hosting abusive images inadvertently. So it is possible for Apple to be, as was said last week, making sure that they're not hosting such images while preserving the privacy of their users. So, you know, I just think it's a cool technology.

I mentioned four researchers at Princeton, one of them being Jonathan Mayer. I hadn't seen his name for a while. We've spoken of him many times and the work he's done in the past. We know that robustly authenticating identity in an online world is difficult. We also know I just invested six years of my life working to create a robust solution for network-oriented or network-based online identity authentication. The standard fallback, unfortunately, is for the agency wishing to identify us to send a text message to the mobile phone associated with our account, the account that we have with them. The problem is this assumes that our identity is tightly bound to our phone, yet of course we are not our phone.

Four days ago, on January 10th, a group of researchers, these four guys at Princeton, published a whitepaper detailing their explicit research, I mean, like they deliberately experimented with how hard this would be to spoof mobile carriers. If their paper weren't coming from Princeton, it might have been titled something like, "Holy Crap, You're Not Going to Believe What We Just Did." But instead their paper carries the sufficiently dry and academic title, "An Empirical Study of Wireless Carrier Authentication for SIM Swaps."

The abstract of the paper, which is perfect, it reads: "We examined the authentication procedures used by five prepaid wireless carriers when a customer attempted to change their SIM card. These procedures are an important line of defense against attackers who seek to hijack victims' phone numbers by posing as the victim and calling the carrier to request that service be transferred to a SIM card the attacker possesses. We found that all five carriers used insecure authentication challenges that could be easily subverted by attackers. We also found that attackers generally only needed to target the most vulnerable authentication challenges because the rest could be bypassed.

"In an anecdotal evaluation of postpaid, as opposed to prepaid accounts at three carriers, presented in Appendix A of this report" - and, by the way, I have a link to their whole report in the show notes. They said: "We also found, very tentatively, that some carriers may have implemented stronger authentication for postpaid accounts than for prepaid accounts. To quantify the downstream effects of these vulnerabilities, we reverse-engineered the authentication policies of over 140 websites that offer phone-based

authentication. We rated the level of vulnerability of users of each website to a SIM swap attack, and we plan to publish our findings as an annotated dataset. Notably, we found 17 websites on which user accounts can be compromised based on a SIM swap alone, in other words, without a password compromise."

So first of all, so as not to keep anyone in suspense, the bad news is that the five mobile carriers found to be vulnerable were AT&T, T-Mobile, Tracfone, US Mobile, and Verizon. In all five cases, the authentication procedures were found to be vulnerable and allowed attackers to readily conduct a SIM-swapping attack.

We should pause for a second, Leo. You and I were just talking about this, what, a couple podcasts ago, how because of the assumption that having the phone in your possession authenticates you, that you don't - and what we learned was that in the U.S., unlike in Europe, in the U.S. SIM exchange can be done by telephone, requiring no physical presence in a store. So you're not having to go to a carrier's physical location, present yourself, show your driver's license or student ID or anything, you know, photo ID, in order to demonstrate who you are. Rather, this is just done over the phone.

And so in the show notes I have a picture of the interaction between a customer service representative and an adversary. So they call up. They phone one of these carriers in this particular flow, claim to be the victim. They request a SIM swap on the account. The customer service representative says, okay, fine. What's the PIN number on the account? The adversary intentionally provides an incorrect PIN. Well, or a PIN. Maybe they get lucky, but probably not. So the customer service representative says, "I'm sorry, sir, that's not the proper PIN." And so they notify the adversary of this authentication failure. And the adversary says, "Oh, crap, maybe I wrote it down wrong. Okay, I guess I don't have my PIN."

So the customer service representative says, "That's quite all right, sir. We would like to know two recently dialed numbers on that phone." So the adversary correctly provides two recently dialed numbers. The customer service representative looks those up and says, oh, yeah, there they are. Perfect. We will fulfill your SIM swap request. And in doing so, the attacker's SIM is then associated with the victim's phone number. All subsequent text messages and phone calls come to the attacker.

And of course then we know, I mean, one of the other things they found was that 17 websites out of the handful that they chose, that's all you need is, oh, sorry you can't log in. We'll send a text message to your phone. What is it? And then of course the text message then goes to the attacker, who types it in. Ah, well, welcome back. What do you want to do?

So here's what they found. There are three key findings. The first is mobile carriers use insecure methods for authenticating SIM swaps. Specifically, one of the things they will ask for is the last payment that was made. They wrote: "We found that authenticating customers via recent payment information is easily exploitable. AT&T, T-Mobile, Tracfone, and Verizon use payment systems that do not require authentication when using a refill card. An attacker could purchase a refill card at a retail store, submit a refill on the victim's account, then request a SIM swap using the known refill amount as the authentication."

Leo: And that's why this is easier on a prepaid account than a postpaid account.

Steve: Right.

Leo: You do buy those cards, yeah.

Steve: Right. So again, a little bit of cleverness. And, frankly, as I was reading these things, these are so bad, but also kind of clever, that I was wishing this hadn't been made public because, you know...

Leo: Why, you want to use them yourself? What? Oh, because anybody could do it, yeah.

Steve: Yeah. I don't want to be a victim, and this sort of makes it like, okay, here's your guide to SIM swapping.

Leo: Yeah.

Steve: Also, second way of verifying somebody, recent numbers. They said: "We also found that using information about recent calls for authentication is exploitable. Typically, CSRs" - and I meant to look up what that is.

Leo: Customer service reps.

Steve: Thank you. "Customer service reps requested information about outgoing calls." They said: "Consider the hypothetical following attack scenario. Using only the victim's name and phone number, our simulated adversary could call the victim and leave a missed call or message that would prompt the victim into returning the call to a number known to the attacker. This call would then appear on the outgoing call log, and the attacker could use it for authentication."

Leo: Which is why I never call anybody on my phone, ever.

Steve: I know. "Customer service reps appeared to also have the discretion to allow authentication with incoming call information, as this occurred..."

Leo: Oh, that's even worse.

Steve: I know, "...four times between AT&T, T-Mobile, and Verizon."

Leo: Don't they know anybody could call a phone?

Steve: Exactly.

Leo: It's crazy.

Steve: I mean, that's how bad this is, Leo.

Leo: Oh, I know, my mom called me. Here's her number. And that would work.

Steve: Yes. "An attacker can trivially generate incoming call records by calling the victim." Duh.

Personal information. They said: "We found that Tracfone and US Mobile allowed personal information to be used for authentication. While our attacker did not use this information, it would likely be readily available to real attackers, for example, data aggregators, and is often public, so it offers little guarantee of the caller's identity. We note that for over a decade FCC rules have prohibited using 'readily available biographical information' to authenticate a customer requesting 'call detail information.'"

Leo: Like mother's maiden name.

Steve: Yeah, exactly. Doesn't matter. Finally - or, no, there's three more. Account information, they said: "We found that AT&T, US Mobile, and Verizon allowed authentication using account information. As with personal information, this information would often be readily available to an adversary. Receipts, whether physical or electronic, for example, routinely include the last four digits of a payment card number. We note that PCI DSS, the industry standard for protecting payment card information, does not designate the last four digits of a payment card as 'cardholder data' or 'sensitive authentication data' subject to security requirements.

"As for the activation date associated with an account, that information can be readily available from business records via a data aggregator, inferable by website or mobile app logs via user-agent logs, or inferable via mobile app API access." They say: "We note that FCC rules also prohibit using 'account information' to authenticate a customer requesting 'call detail information.'" But once again, AT&T, US Mobile, and Verizon all do it.

Device information: "We found that all carriers except T-Mobile use device information for authentication. These authentication methods include the customer's IMEI (device serial number) and the ICCID (SIM serial number). Both the IMEI and ICCID are available to malicious Android apps, and IMEIs are also available to adversaries having any radio equipment." So it goes over the air.

And, finally, security questions: "We found that Tracfone used security questions for authentication. We also found that T-Mobile, Tracfone, and Verizon prompted users to set security questions upon sign-up. Recent research" - as we know - "has demonstrated that security questions are an insecure means of authentication because answers that are memorable are also frequently guessable by attacker."

And then, believe it or not, the second major finding: "Some carriers allow SIM swaps without authentication."

Leo: At all?

Steve: Yes. "Tracfone and US Mobile did not offer any challenges that our simulated attacker could answer correctly. However, customer support representatives at these

carriers allowed us to SIM swap without ever correctly authenticating - six times at Tracfone, three times at US Mobile."

Leo: Well, that's more than a coincidence.

Steve: And I have to say, because I'm approaching 65, I recently needed to sign up for Medicare because I'll be qualifying for that in a few months. That led me on an interesting journey to unblock my credit reports that I'll be talking about in our Miscellany section. And I discovered that just getting mad was all that was required in some cases in order to cause a representative to say, oh.

Leo: Oh, he's mad.

Steve: Okay, sir. I mean, I really wasn't, but it was like, I can't understand what you're saying. What are you asking me? And after doing that a few times, they said, okay, well, never mind. It's fine. And I was just like, what?

Leo: You're old. Whatever.

Steve: Oh, my lord. So, finally: "Some carriers disclose personal information without authentication, including answers to authentication challenges. AT&T, in one instance, disclosed the month of the activation and last payment date and allowed multiple tries at guessing the day. They also guided us in our guess by indicating whether we were getting closer or further from the correct date."

Leo: You're getting warm. You're warm.

Steve: You can't make this up. Oh, no, sorry, you're going further away. Unbelievable. "Tracfone, in one instance, disclosed the service activation and expiration dates. Neither are used for customer authentication at Tracfone." Because in, what was it, three instances, they didn't require any authentication.

Leo: Oh, god.

Steve: "US Mobile, in three instances, disclosed the billing address on the account prior to authentication. In one instance, a portion of the address was leaked. In one, part of the email address was disclosed." Oh, my goodness. "And in three instances the representative disclosed portions of both the billing address and the email." Just to kind of make it easier because, oh, you know, we really do want to help you with this.

Leo: Unbelievable.

Steve: So we're kind of in some deep trouble here, Leo, where as an industry we are clearly relying so much on the possession of our phone, whose communication can rather easily be commandeered and rerouted to an adversary. It's just unbelievable.

Leo: I'm updating Windows. Cumulative update, Patch Tuesday. Retry. There were some problems, it says. Uh-oh. Got to have your Windows update today. This is no day to put it off.

Steve: Actually, I didn't restart this machine. The machine I use Skype with you on Leo is Win10. And because I had time, I let it find the updates. And it gave an error the first time. It wanted me to restart. And I thought no, no, no, no. I'm not restarting. So I just did it, I did a retry, and then it was able to go through the second time.

Leo: That's what I'm doing, then. All right.

Steve: Yeah. Because, you know, I can't restart the same day that I'm doing a podcast with you.

Leo: You don't know how long it'll take to get up.

Steve: Oh, my god. A disaster. So speaking of disasters, this is another one of those, if you, your organization, or anyone you know or care about is using a VPN by the name of Pulse Secure, stop listening right now, yes, and verify that they have patched their VPN server endpoint anytime since last April.

Leo: Yikes

Steve: Of 2019, when an emergency patch to close a serious, let's just say as bad as it gets, remote access vulnerability was patched. So as a consequence, as of the time of this story, nearly 15,000 Pulse Secure VPN server endpoints were not patched. It's heavily used in corporate and government situations. So we're talking nine months after the patch was made available, 15,000 were still vulnerable.

So the story begins, as I said, in April 2019, when Pulse Secure issued an advisory for their apparently aptly named "Zero Trust" VPN product, though I'm sure that's not the way they meant the name to be taken. At the time, they warned organizations of an out-of-cycle patch which fixed a vulnerability in their product known as Pulse Connect Secure. With organizations being as negligent as we know they are with the application even of Windows OS patches, where virtually all the work is done for them, you can imagine how often patching occurs outside of the Windows ecosystem. It's rare. In other words, next to never.

This particular vulnerability is as bad as it gets for a VPN. Our listeners could probably write the next few sentences themselves. It allows people without valid usernames and passwords to remotely connect to the corporate network and turn off multifactor authentication controls, remotely view logs and the cached passwords in plaintext, including Active Directory account passwords in enterprise environments. So it is a full remote authentication bypass for what is apparently a widely deployed corporate and government VPN.

So that was in April. Four months go by. On the 14th of August 2019, someone posted an exploit for the issue on Kevin Beaumont's forum, OpenSecurity.global. Then, a few short days later, a public exploit was dropped by Justin Wagner and Alyssa Herrera. On

August 22nd, with the help of BinaryEdge.io, which is a sort of commercial version of Shodan, which we know Shodan. We often talk about that Internet scanning service. BinaryEdge.io is the same sort of thing.

Kevin determined that someone was actively scanning the Internet for the Pulse Security vulnerability. Kevin said that he sent up a flare that organizations needed to urgently patch this at the time four-month-old critical vulnerability. On August 25th, Bad Packets scanned the Internet and found nearly 15,000 endpoints across the world still having the issue directly exploitable. 5,010 were located in the U.S.; 1,511 in Japan. The U.K. had some, Germany had some, France, the Netherlands, Israel, Switzerland, Canada, South Korea, and then a total of all other countries, 4,052. So all those others were between 800 and 300. So a widespread authentication bypass on a VPN that lets people into the networks behind them.

So those results included networks at governments across the world and many incredibly sensitive organizations, so wrote Kevin, and what was essentially a list of the world's largest companies. So it became clear organizations were not patching. Since then, Bad Packets has been working with the various CERTs around the world, and other security-related groups, to try to get governments to wake up to this and get themselves secured. Kevin works in corporate security, so he follows the ransomware scene and general cyberattacks because he needs to know what attacks are going on and what the attackers are up to.

He had seen the correlation between companies being successfully attacked, that is, with ransomware, and those who used Pulse Secure VPN. And he saw from Internet vulnerability scanning that many of the organizations either had not patched their Pulse Secure system at the time of the incident, or had only patched recently. It turns out that as part of the vulnerability, it's possible to install a backdoor in Pulse Secure systems and gain access subsequently, even after the VPN itself has been patched.

And finally, last week, Kevin spotted two notable incidents where recently breached companies had a strong reason to believe that Pulse Secure was the entry point into their networks for the breach. And in fact it was the Sodinokibi, also known as REvil, ransomware that was installed. In both cases, the organizations had unpatched Pulse Secure systems, and the footprint was the same. Access was gained to the network. Domain admin was then gained. VNC was used to move around the network. They installed VNC via PsExec, masquerading as Java.exe. Endpoint security tools were disabled, and the Sodinokibi ransomware was then pushed to all systems using PsExec. Which, by the way, is the very popular, one of the very popular Sysinternals tools.

Kevin wrote that he had now seen an incident where they can prove Pulse Secure was used to gain access to the network. Today there are still more than a thousand unpatched, wide-open systems on the Internet, despite nine months of pressure to get these things closed. And in Kevin's final update, he added that he had just learned that Travelex - and I had seen this independently - had been hit by a Sodinokibi attack. He noted that Travelex had seven unpatched Pulse Secure VPN servers running.

So I think it's clear that as an industry we still have not solved the problem of communicating the urgency of patching and getting systems patched, especially things like a VPN that are by design exposed to the public Internet. I mean, I've been saying don't let RDP be exposed. Put it behind a VPN. Yeah, but don't put it behind Pulse Secure. Or do put it behind it after being patched. Just, you know, here, 15,000 instances four months after this was made known.

So I don't know how we solve the problem. I mean, and you could imagine each of these is going to have some different story. The guy who was in charge of this got laid off, and the Pulse Secure account had his email address that was then discontinued because it

was gone. And so the notifications that they may have sent, I don't know if they did, but assuming that they were as responsible as they could be, they would have sent out notifications to everybody that they knew were using their VPN. Well, it just bounced. So what could they do? Somehow this problem needs to get solved. It's clearly still a big problem for our industry.

Oh, and speaking of patching right away, I will have a little bit of good news about Firefox in a minute. In this case, it was a brief hiccup. China's Qihoo 360, the security company that we've often mentioned, spotted a serious, as in CRITICAL in all caps, type confusion bug in Firefox's IonMonkey JavaScript JIT - we know that stands for just-in-time - compiler which was being abused in the wild. And as we know, that makes it a zero-day vulnerability. Just two days after Mozilla released Firefox 72, they issued an immediate update to patch and resolve this critical zero-day flaw.

As it happens, I was using Firefox over the weekend, and my Firefox said, hey, give us permission. We need to restart right now. And I said, oh, okay, and did. And I got 72.0.1, which is the now updated with this just-in-time compiler zero-day flaw fixed. And what's interesting about this is that attacking a just-in-time compiler is both clever and common because, as we know, compilers like interpreters tend to be a bit finicky.

But also, in the special case of a just-in-time compiler, it's their job to create new executable instructions on the fly. That means they cannot be constrained by the usual safeguards provided by DEP, Data Execution Protection, which would otherwise prevent the execution of data, because executing freshly compiled data as instructions is precisely what JIT compilers must do. So it's sort of an interesting aspect of just-in-time compilation that, by their nature, they're not able to have the same level of rigorous protection that statically compiled code is able to provide.

Qihoo 360 notified Mozilla of the trouble, indicating that they discovered the flaw after observing targeted attacks occurring in the wild. Mozilla instantly fixed it and pushed out a fix. So anyway, everybody should be using 72.0.1. And, you know, I've seen Firefox waiting to be asked. Maybe they're not because this is a critical problem, and so it'll be a little more aggressive. But I've often gone to the Help>About Firefox to see what I'm running, and that woke it up to the availability of an update, which I then of course immediately allowed it to apply. So it might be worth, if you're a Firefox user, just going under Help>About Firefox and making sure that you have 72.0.1. And if not, you'll get it right then.

Leo: Oh, it won't download it and then just have it, and then if you restart? Because I know it can't obviously apply it when it's open. If you quit it and open it - see, most people never quit their browser. That's part of the problem.

Steve: I know, that's the way now I'm just - it lives over on my lower left screen. It's like my portal to the Internet, yeah. And so I don't think...

Leo: Restarting your machine isn't a bad idea either once in a while; right?

Steve: That's a good thing to do, too.

Leo: The whole thing; right? Yeah.

Steve: It does, it would flush any little RAM critters out that might have crawled in.

Leo: RAM critters. Hate that when it happens.

Steve: Speaking of crawling in, I had to crawl out, Leo.

Leo: Uh-oh.

Steve: Of a movie theatre on Friday.

Leo: Yeah, so you didn't like, what is it, DX? You went to see what DX? 4DX?

Steve: 4DX.

Leo: What's that?

Steve: I didn't realize what I was getting myself into. And I would never have imagined that I would walk out of a Star Wars movie. Now, okay. Bar Bar Jinks or whatever...

Leo: Oh, I know what this is. Jar Jar, yeah, yeah. No, no.

Steve: Jar Jar, god.

Leo: This is with the moving chair.

Steve: Oh, my god.

Leo: We have these in Petaluma. They don't call it 4DX. They have another name for it.

Steve: Oh, well, that - oh.

Leo: This one looks like it moves more than ours does. Ours is just a little bit of movement.

Steve: My first indication that perhaps I'd made a serious mistake in choosing the theatre was when I spotted the control on the arm rest to turn off being sprayed with water.

Leo: Okay.

Steve: I'm not kidding.

Leo: Yeah, ours doesn't do that.

Steve: There's a button on the arm rest whether you want water spray or not.

Leo: Oh, my god.

Steve: And that was - I was a little concerned. The chairs were in sets of four. And there was all this extraneous stuff lining the edge of the theatre on both sides.

Leo: Including apparently water spritzers.

Steve: It was unbelievably wrong.

Leo: Even the guy in the ad doesn't look that happy about it.

Steve: No. He's gripping his cup holders on either side for dear life.

Leo: Help me. Help me.

Steve: Lorrie ended up having to kind of sit forward because the seat was just throwing us around so much. And, I mean, she's no weenie. She, like, her comment was, "The ones at Disney are done right," where if you're going on the roller coaster over the edge, you dip forward. I mean, for example, we're looking at the Millennium Falcon dodging and jumping and things. It's in the distance, and we're being thrown around in our seat.

Leo: Uh-oh, yeah.

Steve: Even though it's not like we're in the Millennium Falcon and we're seeing the star field out the front. We're, like, passive observers. And there's strobes flashing on the side. There's clouds of mist coming up from both sides of the screen. It's the most - it is so ridiculous and wrong.

Leo: Didn't you suspect something when you saw that the ticket price was $26.70 each?

Steve: That should have been my clue. The good news is we left after about 30 minutes. Finally my buddy Mark kind of looked at me, and he said, "Are you kidding me?" And I said, "Let's just get out of here." And so...

Leo: It's a shame because it spoiled the movie for you. I mean, it's...

Steve: Oh, Leo, it threw you, almost literally, out of the experience. I mean, you couldn't pay attention. And I do have to say that what I saw of the movie wasn't that impressive. It looked like a whole bunch of frenetic action just for its own sake. On the other hand, this had to be an incredibly expensive installation. Mark was commenting, he said, "Did you see the people trying to get into their seats?" Apparently someone showed up late, and they were like, their popcorn was flying in the air because they were trying to sit down, and this thing was jumping around all over the place. Oh, my god.

So, oh, and the other thing, the other problem was it was in 3D on top of it being in, like, 4D. And I looked, I couldn't believe the glasses were red/green tint. They weren't the RealD 3D that we talked about years ago which use a really cool clockwise and counterclockwise polarization. This was red and green 3D. I thought, what year is this? Anyway, enough said. Just a little heads-up for any listeners who want to go see Star Wars. And Lorrie said, you know, this is for 13 year olds. Maybe.

Leo: Well, Star Wars is for 13 year olds, too. But we have - you went to 4DX. We have D-Box in Petaluma. But it doesn't spit at you. It just does a little bit of this.

Steve: Apparently there are six different smells that it can also...

Leo: Oh, I don't want smells. No, thanks.

Steve: I'm not kidding you. I'm not kidding.

Leo: The real problem is, as you can tell, the moviemakers don't make the effects. The 4DX guys make the effects, right, after the fact.

Steve: And they're trying to sell how wonderful it is that you could just, you know, you bring a martini, it'll be shaken and not stirred.

Leo: They should have really said you have to be this young to ride this ride. I don't think they should have let you in.

Steve: The good news is there was no complaint about getting our money back. I said, "Come on, this is ridiculous." And this sounded like it's not the first time that it happened, either.

Leo: Yeah. Yeah, they knew. But do go, you know what, I like the movie. You can wait till it gets, you know, watch it at home. But I liked it, only because it completes the adventure. It's the end of the nine-movie series.

Steve: Yes. And I do have to see it. We did enjoy "The Mandalorian." And because my Disney subscription is up on the 24th, last night I just sort of browsed through to see if there was anything else I wanted to see there, and we watched the first half of the

making of the trilogy. Which I really - we ended up stopping halfway because it was bedtime. But we're going to absolutely finish it. It was really fun. It's two hours long, and it's the inside story of how the original trilogy got made and almost didn't get made, and all the things that went wrong. And anyway, so if anyone hasn't seen it, I would say it's probably worth finding somewhere.

Leo: I have Disney Plus. I've yet to watch one thing on it. So okay, now I'm going to watch that, I guess.

Steve: Well, and the only thing that Lorrie thinks "The Mandalorian" was worth was Baby Yoda.

Leo: Yeah.

Steve: You know, everybody thinks is wonderful.

Leo: Hit with the kids.

Steve: So I don't think I'm going to continue.

Leo: Yeah.

Steve: And I mentioned applying for the first time for - I needed an account at Social Security in order to sign up for Medicare. And I immediately got blocked when I was trying to create my Social Security account, and I was told to call a toll-free number. So I thought, okay. And so I called. The first question I was asked was "Have you locked your credit bureaus?" And I said, "Yes, I have." And she said, "Well, that's why we cannot process you electronically. Here's a code. Go take it to your local Social Security office in order to proceed." So I did that. And that was all fine. But so, and the point was, I said to her, I said, "Well, yeah, I locked it to prevent identity theft." And she says, "Yes, well, it's working."

Leo: Yes, it is. Yes, it is.

Steve: And I said, "Oh, that's cool." So but the other thing that...

Leo: I have two-factor on my Social Security account. But guess what the second factor is? A text to my phone.

Steve: Uh-huh. Yeah. Yeah.

Leo: I guess, you know, what are you going to do?

Steve: So the other thing I decided to pull the trigger on, since basically everything I buy is through Amazon, is the Amazon store card because, as a Prime member, you get 5% off across the board. And it's nuts that I'm not getting 5% off of pretty much everything I buy.

Leo: True.

Steve: So in order to qualify, because my credit is locked - and I'm telling everybody this because long ago we talked about locking Equifax, Experian, and TransUnion. I just wanted to share my experience, which was that I was able to unlock all three. There is no charge for doing it. And all three now allow you to do a transient unlock, a temporary unlock.

Leo: That's great.

Steve: Which is what I - yeah, it's very nice. I was able to say that I wanted my queries to my reports allowed until January 10th, and that allowed me a few weeks of time for Amazon to pursue checking my accounts and so forth. And it all worked. So each of them - I have in the show notes, if anyone is interested, I have three toll-free numbers for Equifax, Experian, and TransUnion, which are what you call if you want to do a temporary unlock of your credit. And you're able to give them a date where you want it then to automatically relock. And there's no charge for this. So things have improved since the days that we first talked about this.

Leo: They improved because Congress made them.

Steve: Yes, I know.

Leo: They didn't improve on their own.

Steve: And I'm thankful, thankful for that.

Leo: But they used to charge for locks. They used to charge for unlocks. They used to charge in some states as much as $40 to lock and unlock, and Congress finally said, no, you can't charge a thing to do either.

Steve: That is so wrong.

Leo: And so, yeah, it was terrible, yeah. So lock. There's no penalty to locking. And I like the transient unlock. I think that's great.

Steve: Yes. And I wanted to tell our listeners again, I mean, I'm sure everybody has heard the horror stories about identity theft. Identify theft occurs when somebody typically applies for credit in your name. And then the credit grantor checks your credit in

order to grant the credit. Now somebody who's not you has the ability to incur debt that is essentially yours.

Anyway, so the point is, if you're a person who's actively needing to get credit, then it's probably an inconvenience to lock things down. But if you're like a lot of the listeners of this podcast, and you and I, Leo, where we're sort of past the age where we're applying for credit, you know, we're not newlyweds with kids and buying things...

Leo: Only because you never asked me, Steve.

Steve: I really think locking one's credit is a slam dunk. It's a no-brainer.

Leo: Just understand that that means you can't get credit. You can't buy a car. You can't, you know, you can't get a car loan. You can't get a house loan. You can't get a credit card. That's what you want. And it's nice that they give you this transient unlock so that you can do those things and then go back to a secure state. I think it's really good, yeah.

Steve: So I was in the process of sort of looking at my domain names, GRC's domain names, because I've got a lot of things that are something.grc.com - www.grc, news.grc, SQRL.grc, blog.grc and so forth. And EV certificates cannot have wildcards. And as we've been talking about recently, EV certs are kind of like, eh, okay, well, the browsers aren't showing them anymore, so what's the point? And given that you have to have a bunch of EV certs or a multidomain EV cert, that is, EV certs don't allow you to use wildcards, I'm seriously considering switching back to, you know, I do not want to do Let's Encrypt. It's not often talked about, but Let's Encrypt and any of the ACME-based automated cert issuers are really being used and abused now to the point where I wouldn't be surprised if at some point browsers put up a little indicator to indicate that, yes, you have a secure connection, but this is from an automated certificate, not one that has had a human in the loop.

I'm going to keep using DigiCert because they're my Certificate Authority, and I have no problem with satisfying them that I am me in return for having a certificate that a bot did not issue. I mean, I get it. There's places for bots. But I don't think that's the case for certainly a security-oriented ongoing enterprise.

Anyway, the point is grc.sc is one of the domains. That's my little shortcut domain. And something, one of the other domains I have is grctech.com. I have grctech.com, which I created in order to have a third-party domain not under GRC, in order to check cookie handling in browsers. And this is years ago. And so I was thinking, yeah, you know, maybe it's just time to shut that whole thing down, that I don't have to bother with grctech.com. I thought, I'll just take a look in on the cookie-handling stuff.

So I went to - and I'm going to recommend our users do, because what I found was surprising. I created a new shortcut, grc.sc/cookies. I first ran it in Firefox because that's my default go-to browser. And it went through the test. Grc.sc/cookies does an extremely comprehensive test of your browser's current cookie-handling settings. Came up perfectly in Firefox.

And I'm not sure what caused me to check it in Chrome. But I did. And Chrome has a problem with cookie handling, which I never knew. It does not properly handle first-party persistent cookies. And the page that came up, my own page, demonstrated the fact, and it explains it in English, what's going on. The page explained this browser's exchange

of first-party persistent cookies is enabled, and some cookies are being freshly exchanged. Some anomalous cookies are also present, so please see the additional points below.

And then down below I explain: "The first-party persistent cookies shown above as stale, in orange, was previously accepted by your browser. An updated fresh cookie was just offered to this browser, as indicated by the cookie's label being black, but this browser ignored the updated cookie and instead returned the stale one it already had." And I happen to know because it shows that was 182 days, 21 minutes, and 7 seconds old at this time.

Anyway, Chrome has a problem with the first-party persistent handling of cookies set on icons, which may or may not be a problem. Anyway, I just thought it was interesting. I wanted to bring it to our listeners' attention. Grc.sc/cookies will take you to GRC's Cookie Forensics page. And I think I'm going to leave it up and leave it running and renew the grctech.com domain because it looks like it's still useful. I had assumed all browsers had fixed this. Back then, when I initially created it, there were all kinds of problems with browsers and handling cookies wrong for different types of content. They'd been fixed, I thought. But not so for Chrome.

Leo: You know, it's funny because it's today the story came out that Google says it's going to phase out support for third-party cookies in Chrome within two years. They won't have it at all.

Steve: No kidding.

Leo: It won't even be in the browser.

Steve: Whoa.

Leo: Yeah.

Steve: I did not hear that. Hallelujah.

Leo: Yeah.

Steve: Wow, that's going to change the world.

Leo: They're also, weirdly, phasing out support for user-agent strings. I don't know...

Steve: They're just not going to do it?

Leo: Apparently. They're going to do something called "client hints." This is all - in fact, I'd love to get at some point a show about this - part of their new privacy sandbox project. And so they're doing a lot of things to - which is weird because

Google, of course, an advertising company. But I think they see the writing on the wall. So if they're not to lose market share to Firefox and Brave and everybody else, Edge, they're going to have to do something about Chrome. And so they're changing a lot of things - user-agent strings and third-party cookies.

Steve: Good for them.

Leo: Yeah, yeah. It's interesting. I don't, you know, I think that, well, I'm always skeptical when Google says, oh, we're going to protect your privacy. But in this case it kind of sounds like they're going to. Oh, yeah. We're all about privacy, yeah.

Steve: So before we take our final break, I'll just note, as we said at the top of the show, that today is the final day for Windows updates.

Leo: Get them. Windows 7 updates. Yeah, get them.

Steve: Sorry, Windows 7 updates. And presumably, well, and we know, as you said, Leo, we believe this Crypt32 API in Windows needs to be fixed. So it's being fixed, so that's good. I've pretty much made peace with Windows 10. That's what Lorrie's running. I'm about to rebuild my "A" system, which I've just been using a closed Lenovo, the Carbon X, which I like a lot. But my tech support guy, Greg, has been saying that his laptop is really old. So I thought, you know, I never really use it. I'm going to give it to him. So I'm going to rebuild a system for myself based on an Intel NUC, which I really like those. That's what I built for Lorrie, and I'm really happy with it. And I'm going to set up Windows 10.

So Windows 10 will be one of my main go-to platforms. You know, you have to strip all the crap out of it and spend some time making it usable. There's no way I'm having Candy Crush Soda Saga on my menu. But it can be fixed. So I just sort of wanted to set the record or update the record for where I stand. I'm sitting in front of a Windows 7 machine which will stop getting updates after today. And it's such a pain in the butt to set up a new system from scratch that, since I'm having to do it in one case already, I guess I could clone that.

Leo: Yeah.

Steve: Maybe I'll do that.

Leo: Except that it is pretty specific to the hardware.

Steve: Yeah. Well, no, I would clone it with another NUC. I actually have two identical. I deliberately got the older NUC, the one that could still run Windows 7, because the newer one won't at all, when I thought I was going to end up using Windows 7 on it. But I'm going to end up using Windows 10. Because, again, I don't want to be stuck in the mud forever. It doesn't make any sense. Oh, and when I was looking at grc.sc, I'm seeing that the Windows 10 upgrade shortcut that I created is still getting a lot of use. So I thought I would remind our listeners, grc.sc/win10 - W-I-N-1-0. Yup, grc.sc/win10. That just bounces you over to the link at Microsoft where you can download Windows 10 and

install it on top of your Windows 7 system, and then you'll be getting upgrades or updates next month.

Leo: Good. All right. Let's talk about Cable Haunt.

Steve: So some Dutch researchers discovered this. As a consequence they checked Europe and found more than 200 million Broadcom chipset-based cable modems vulnerable to what they found. I would imagine that probably means worldwide, what, half a billion? That is to say, more than 500 million? The website is CableHaunt, C-A-B-L-E-H-A-U-N-T, dot com. And I have that link and a link to the report, their full PDF report in the show notes.

So let me explain what new horror we have here, and what it means. Cable Haunt is the name given to a new critical vulnerability found in cable modems from various manufacturers around the world. The reason so many various brands share the same problem is the very worrisome tendency we're seeing toward a monoculture. Broadcom is by far the dominant supplier of cable modem core technology. And Broadcom published some reference firmware which everyone copied. Remember we saw this in the Universal Plug and Play, UPnP, where Intel published something that wasn't meant to be used, and everyone just said, oh, it works.

Leo: This is an example, folks. This is how you would do it maybe sort of. But don't forget you've got to put error checking in.

Steve: Yeah, wouldn't that be nice. So this vulnerability ultimately enables remote attackers to execute arbitrary code on vulnerable cable modems, and pretty much everyone's cable modem is vulnerable. At this point it looks pretty much like all modems are vulnerable. There are some, I'll mention some, that look like they're older, that are based on a TI chipset. But, I mean, like my cable modem is vulnerable. Looks like they all are, with exceptions. So this exploitation is accomplished indirectly through an endpoint on the internal local network.

Leo: So they have to be inside the house.

Steve: Well, they have to be on your browser. Your browser can do it.

Leo: Ah. So malware can do it, yeah.

Steve: Malware could do it. A bad ad could do it. A compromised IoT camera. Anything on your LAN is able to do this. Through this malicious communication a buffer overflow could be exploited to gain control over the cable modem. So the researchers write: "There are an estimated 200 million cable modems in Europe alone. With almost no cable modem tested being secure without a firmware update, the number of modems initially vulnerable in Europe is estimated to be close to 200 million."

Leo: Wow.

Steve: "However, it is difficult to give a precise estimate of the reach of Cable Haunt." And of course everybody's busy testing their modems. There is a Linux script, a Linux Python script available, so Linux users can test. Unfortunately, it doesn't look like there's an easy way to do it in Windows. I'm hoping, you know, it's the kind of thing I would normally do one of my jiffy quickie things. But everyone wants me to get back to SpinRite 6.1 - and I do, too - so I'm not going to do it. And the other thing is it's just not that difficult. I'm sure this time next week I will be talking about some freeware that has been created to allow people to check their environments. There are some things you can do immediately that are cool, that I will share during this podcast.

So the reason, they said: "The reason for this is that the vulnerability originated in reference software, which has seemingly been copied by different cable modem manufacturers when creating their cable modem firmware. This means," they're saying, "that we have not been able to track the exact spread of the vulnerability and that it might present itself in slightly different ways for different manufacturers. We have contacted as many of the largest ISPs" - and they're talking Europe - "and manufacturers as we could ahead of time, to give them time to fix the issue, but with varying success. Some of the contacted ISPs have informed us that they have or are rolling out firmware updates. However, we're still missing updates from several, and some have wished not to be listed on this website. The ISPs that have confirmed their modems are secure can be found below." And this is in their report. And actually their website has a bunch of FAQ expandable tab things down at the bottom where there's a lot of additional information.

"The affected component is the cable modem's core OS, which is eCos" - e-C-o-s, which I had seen before. It's a widely popular embedded multithreaded real-time OS. Being an embedded OS, it's meant to be small and fast and lightweight. And since it also only runs its own trusted compiled-in code, it completely lacks any of the now common anti-malware preventions such as address space layout randomization, protections against stack smashing or execution and other mitigations. Thus it freely allows execution of code on the stack. That is, it never expected to have a problem, so it doesn't check for it.

Leo: I've just checked. All three of my cable - four of my cable modems are vulnerable. My Arris and my Netgear, yeah.

Steve: I know.

Leo: But I run that myself. My ISP doesn't. I got my own. So does that mean - my ISP can't fix it. I have to fix it; right?

Steve: No. I also got my own. I have a Netgear CM1000, a DOCSIS...

Leo: That's what I have, yup.

Steve: Yup, a very nice DOCSIS 3.0. I know because I have a friend at Cox, thanks to this podcast, who updated my firmware on that. And it is not the case that a subscriber is able to apply firmware themselves. Only the WAN side, only your ISP is able to update the cable modem. They refuse to have foreign firmware attached to their network is the way they think of it.

Leo: But if you bought it yourself, it's still foreign firmware if you update it. Only the cable company can do it.

Steve: Yes. You're unable to update the firmware on your cable modem.

Leo: Now, some ISPs are doing an update. Eric in our chatroom lives in Sweden. His Swedish ISP, Com Hem, did a security patch for Cable Haunt. It took out 300,000 customers for an hour.

Steve: Yes. That's going to happen.

Leo: Oh, my god.

Steve: But on the other hand, that's good news because, well, just wait till you hear how bad this is. So there's no protection. The result is that exploits are unusually easy to write, to implement, and to run with high reliability. Code is always at a fixed known address since there's no randomization. And the OS never performs any stack sanity checking, making it entirely vulnerable to buffer overflow attacks. This made finding the Cable Haunt exploit easier for them and makes it that much easier to be exploited.

Leo: The irony is that code was probably written in the room I am sitting in right now. This used to be a Broadcom facility.

Steve: Now, get this. Our cable modems all have, and I never knew this, I ran it this morning, a built-in spectrum analyzer which can be used to identify connection troubles with the cable system. Although the port which exposes the spectrum analyzer may vary by modem make and model, it is readily discoverable with any port mapping tool such as Nmap. And the proof-of-concept Python script performs this port scan to locate the spectrum analyzer.

Now, first of all, there is a very cool port scanner which I used this morning that I will recommend. I used GRC's shortcut. It's grc.sc/aps - advanced port scanner - grc.sc/aps. It's free. Does not require installation. You can install it if you want to. Nice-looking gal on the website home page. They produce advanced-port-scanner.com and also advanced-ip-scanner.com. Both are free. Both are cool. The IP scanner, I ran it, and it found all of the systems that I have installed on my LAN across the entire IP space.

Leo: It's Windows only, unfortunately.

Steve: It is. But there are, you know, LAN port scanners are a dime a dozen now. You could easily find one for Linux and for Mac. So if you just google, like macOS port scanner, you know, or LAN port scanner, I'm sure you'll find a good one.

Leo: Or you can even do it from a command line; right? I mean...

Steve: Yes. Yeah, yeah. This one showed me, when I ran it, that port 80 was open on my cable modem, and port 8080. But I'm getting ahead of myself. So request to the port scanner on - and the IP address for my cable modem is 192.168.100.1.

Leo: Macintosh comes with a network utility app that has a port scanner built in.

Steve: Ah, very cool.

Leo: So you don't have to use Nmap. Nmap will do it, but, yeah, it's built into macOS; yeah.

Steve: Yeah, nice. So my cable modem is at 192.168.100.1. And it turns out it's listening for its normal web connection on port 80. But it has a previously, unknown to me, spectrum analyzer that you can bring up with your web browser by going to that IP:8080.

Leo: How do you - that's not your router? That's your cable modem? It has its own address?

Steve: Yeah, yeah.

Leo: Your router also has its own address. Don't confuse the two, obviously.

Steve: Correct. Yeah, the router will be normally 192.168.0.1 or .1.1 is generally where today's routers live. This thing is .100.1.

Leo: Got it, okay.

Steve: And if you put that address into your web browser, you generally go to that router. And it'll say, you know, log in. And you're able to look at some normal stuff like the number of corrected and uncorrected packets, how many errors it has. You're able to spot - it's sort of interesting, if you've never done it before.

Requests to port 8080 are sent JSON formatted through a WebSocket. A WebSocket is a JavaScript-y way of initiating a TCP or UDP connection over the Internet. However, the JSON deserializer inside the cable modem allocates - get this, Leo - a predefined amount of memory for each JSON parameter.

Leo: Oh, I can see the problem already.

Steve: Uh-huh. It will keep reading input parameters.

Leo: Sure, as long as you give them to it.

Steve: Until a comma is reached in the input stream. Not surprisingly, this can be easily exploited by a malicious request. In their example, and I show them here in the show notes, the fStartHz, you know, the frequency, the starting frequency for the spectrum analysis, fStartHz parameter, has a larger value than was allocated in memory and will therefore overflow and overwrite the registers. They said: "To validate this, a JSON package with 200 A's as the fStartHz parameter can be sent through the serial connection to the cable modem. This will crash the modem, and all register values will be displayed, showing that the program counter has changed to 0x41414141," which we all know is four capital A's in hex. And so you can see in the show notes a JSON-formatted query and where they changed fStartHz to all capital A's.

The eCos OS saves the caller's registers, when a subroutine is called, saves the caller's registers on the stack and restores these before returning. Therefore, if the variable registers S0 through S7 are overwritten, and the return address register is saved on the stack, as is the case, it's trivial to run any existing code in the system with the attacker's desired input variables. They said in the report, although they did not bother to engineer the execution of their own code, they used Return Oriented Programming (ROP) to essentially execute any existing code on the system in what they describe as a Turing-complete manner, meaning they were able to use existing code just before return instructions to get anything done that they needed to get done, manipulating the system extensively.

This, then, they used to open a telnet server for external root access to the cable modem, allowing remote access to the entire system using the cable modem as a telnet server out to the public Internet. Through this telnet connection they were then able to access a range of methods, including reading and writing arbitrary memory addresses, executing code from any memory address, including ones just written to. They noted that the last steps vary from modem to modem, but they provide a complete example in Appendix B of their report.

So the attack can be executed by having the victim run malicious JavaScript. They said a common avenue of attack would be a link that is opened in a browser, but could for example also be done through ads on a trusted website or an email client. And as I noted, Leo, anything on your network has access out to the cable modem on your perimeter, so if a webcam got compromised, or any IoT device.

Anyway, so it turns out that the JavaScript running in the browser establishes a WebSocket connection directly to the modem through the local IP address. Normally the WebSocket access would be security restricted, but it's up to the server to enforce the restriction, and they didn't implement that in the cable modem because we're all friendly. This is on the LAN side. Nothing malicious would ever happen. And besides, who cares if you see a spectrum analysis?

What they have verified is that it is possible to change the default DNS server; to conduct remote man-in-the-middle attacks; to hot swap code or even entire firmware; to upload, flash, and upgrade the firmware silently; to disable the ISP's subsequent ability to remotely upgrade firmware in the cable modem; to change any config file and settings; to change associated MAC addresses; to change serial numbers; and to host a botnet.

So as we mentioned, unlike our routers, consumers do not update our own cable modems. This is only done from the broadband WAN side. And as I mentioned, thanks to this podcast I have a friend at Cox in Atlanta who's deep into the technology. And in the past he has pushed for more updates to my Cox-connected modems. So I'm sure it can be done. It does cause an outage while the modem receives the firmware, shuts down, reboots, then relocks to the network. But I think it's entirely foreseeable that pretty much everyone who has a cable modem is going to be seeing a brief outage.

Leo: You can see why, though, an ISP is going to be reluctant to do that. I mean...

Steve: Oh, lord, yes. Because who knows...

Leo: The calls they're going to get, it's going to cost them a lot of money. I mean, Comcast is so huge, they're going to get a million support calls.

Steve: Yup.

Leo: My cable went out. My modem went out.

Steve: Well, and of course they've been pushing phone. So now phone service will be out.

Leo: Oh, that's right. Anything Internet.

Steve: TV service, yes, TV service will go out.

Leo: Not cable TV, but anything over your modem will.

Steve: Oh, you're right, you're right. Cable modem, right. So the digital phone service will go out.

Leo: If you're watching Netflix, bye-bye.

Steve: Yup.

Leo: Not that Comcast minds about that.

Steve: And I'm sure they'll do it at 3:00 a.m. so as to minimize the disruption to everyone. But still, a lot of people have stuff, I mean, who knows, you know, streaming things, and they're just assuming that they've got their Internet up. And when they find out it goes down, they'll have no idea why because there's no way for Comcast to notify anybody.

Leo: Right.

Steve: So Threatpost updated their earlier coverage this morning by adding: "As far as U.S. ISPs are concerned, a Cox spokesperson told Threatpost, 'We are rapidly testing all our in-home broadband equipment, determining any vulnerability and the best steps to mitigate as needed.'" And a spokesperson with Charter told Threatpost that Charter is

"currently working with each of our vendors to determine if their equipment is vulnerable and when we could expect to see a firmware upgrade."

Anyway, so as I mentioned, I ran the Advanced Port Scanner, grc.sc/aps, or just google "advanced port scanner." It's a nice little bit of freeware with a good reputation.

Leo: And as I mentioned, Mac people have something built in, in the network utility. And at least my install, which is based on Ubuntu, and I bet you all Linux installs, netcat is installed. So you probably already have that in Linux, as well.

Steve: Nice.

Leo: Netcat.

Steve: So do that against the IP for your cable modem.

Leo: How do you figure out what that is?

Steve: That's a good question. I knew that mine was 192.168.100.1. It is in the manual for your cable modem because there is an admin interface that wants you to change from the default.

Leo: Oh, okay.

Steve: In some cases that will prevent the attack, although I believe, and this is spelled out in the website, for the modems they know of, it looks like there is a ready authentication bypass, as well. But Leo, scroll down. Look what I found. I discovered this in my cable modem. At :8080 - oh, what was really interesting, it was cool that - go a little bit further down. When I went there under Firefox, what I got was "Spectrum Analyzer not supported in this browser. Please use Safari or Chrome." Which again, remember that the people who discovered the attack realized that using Firefox made you safe. Firefox cannot be used as the jumping-off point for this. On the other hand, remember that anything in your network can be. So this needs to get fixed.

Anyway, there's a spectrum analyzer. And it was, like, updating in real-time. All this was like going "blump blump blump" and changing. It was just amazing. I had no idea that was in there. So because I use pfSense, and I have a firewall between my LAN and my router, I installed a rule to block access to port 8080 at that IP. So until Cox gets around to updating me, I'm secure. And so that is certainly a short-term workaround that users who have the ability to add some firewall rules to their router could employ in order to...

Leo: Say that again. So if you block port 8080 inbound, that will...

Steve: 8080 for me. They say that the port...

Leo: Whatever your spectrum analyzer port is.

Steve: Yes, exactly. And that's why you need the port scanner in order to for sure locate which is the port where that's operating.

Leo: Somebody in the chatroom says, am I not vulnerable if I don't have a spectrum analyzer on my cable modem?

Steve: That's a good question. You're not vulnerable if you don't have Broadcom. And apparently all Broadcom modems do have the spectrum analyzer.

Leo: Have the spectrum analyzer, okay, all right.

Steve: Yeah.

Leo: And I was looking at the list. It's all the commonly used modems.

Steve: I know.

Leo: And presumably Comcast has to be able to ping my modem and say what are you so it can apply the appropriate patch. They can do that? DOCSIS 3 will let you do that?

Steve: Yes.

Leo: Yeah, okay.

Steve: Yes. They absolutely have the ability. In fact, my friend in Atlanta was able to put my cable modem under observation for a while. In fact, this all happened, Leo, when we were having those dropouts on my connection. I was able to get some amazing service, thanks to the podcast.

Leo: By the way, it's happening right now.

Steve: Oh.

Leo: And if you're doing a podcast, do not patch your modem in the middle of a podcast, also.

Steve: Also, Tom's Hardware has some coverage. They said the Lyrebirds researchers say models known to be vulnerable include the Arris Surfboard...

Leo: That's the one I have.

Steve: ...CM8200A, Arris Surfboard SB6183...

Leo: That's the one I have, yeah.

Steve: ...Arris Surfboard SB8200, COMPAL 8284E, COMPAL 8486E, Humax HGB10R-02, Netgear CS3250EMR, Netgear CG3700EMR, Netgear CM1000.

Leo: That's what I've got.

Steve: You and I both have those also. That's what I have. The Sagemcom F@st 3686, the Sagemcom F@st 3890, Technicolor TC4400, Technicolor TC7230, and Technicolor TC7300, although some firmware versions of those models may not be at risk. And they talked about, oh, it said: "We discovered that our aging Arris Surfboard SB6141 uses a TI chipset, so we're out of the woods. But two later Arris models, the Surfboard SB6183 and 8200, do use Broadcom chipsets, and the latter is on the list of known models vulnerable to Cable Haunt." So essentially, recent modems, and pretty much every modem.

So again, the reason I think this is worth thinking about is, I mean, like paying some attention to for our listeners, is that this is, like, hundreds of millions of cable modems. The good news is they can be responsibly patched without end users needing to do anything. The bad news is it's going to take a while for, I mean, and I don't know how long. Weeks? I mean, they're going to be in a hurry. The cable companies are going to certainly be on this. Broadcom has been asked about this. They said that - this might have been Tom's Hardware. Yeah, it was: "We've reached out to Broadcom for comment, and a company spokesperson gave us this statement: 'We made the relevant fix to the reference code, and this fix was made available to customers in May of 2019.'" So there's another thing that's annoying.

Leo: What? What?

Steve: Yes. This has been known and been available to all of our ISPs since May of last year. And now, only because it came to light, they're all running around and scrambling. So once again it's like, oh, well, maybe it's not a problem. Uh-huh.

Leo: Well, I know why they don't want to do anything about it. This is going to be a big hassle for them. They're knocking people's Internet out.

Steve: Yup.

Leo: For at least some time, the time it takes to reboot.

Steve: Yup.

Leo: There is a list also of vulnerable modems on the Cable Haunt site, too. It looks like it's pretty similar to the Tom's Hardware list.

Steve: That's probably where they got it, yeah.

Leo: Yeah. Oh.

Steve: Yeah.

Leo: And the thing that's frustrating is there's nothing you can do about it. You have to just get your ISP to do something about it.

Steve: Correct. The end user is unable to update their firmware. It's only doable from the WAN side.

Leo: Although there is this kind of weird fix where you block the spectrum analyzer port. If you could figure out with a port scan which port it is, you might then use a firewall rule to block that port. Where would the firewall, though, wouldn't that be on the router inside the cable modem? Where would you - does the cable modem have a firewall, too?

Steve: Well, so if you have a cable modem, normally that goes then...

Leo: Oh, I see. I get it. Because it's coming from inside the house.

Steve: Exactly.

Leo: So your firewall rule is not from outbound traffic coming into your cable modem. It's from your computer going to your cable modem.

Steve: Right. Right.

Leo: So you want to block outbound traffic on that port.

Steve: Yes. You want to block your router sending something to what it sees as the WAN, to 192.168.100.1, port whatever it is.

Leo: So that's a little bit of research for most of us because we have to figure out, A, what our cable modem address is, and then what port the spectrum analyzer uses.

Steve: Right, right.

Leo: Wow. What a mess.

Steve: So some homework for all of our listeners.

Leo: Yes.

Steve: It is a real mess. It is, again, it's this monoculture is a problem. If something big like this is discovered, and everybody is using the same one, I mean, it's potentially devastating. Whereas if we had a much more heterogeneous environment, it'd be like, well, yeah, some people. But my point is that there's no way that hackers are not on this right now. I mean, there's just no way that they're not working to come up with an ad that they will stick into the ad stream that will be delivered to people's browsers, most of which are now Chrome, and will then attempt to access their cable modem, and can right now.

Leo: Nice.

Steve: Yeah.

Leo: I bet you it's not fixed for some time. I'm just guessing.

Steve: That's my feeling, too. It just feels like, you know, they've had it since May.

Leo: Since May.

Steve: Yup.

Leo: Criminently. Well, thank goodness you listen to this show. That's all I can say. One more reason that you cannot miss a Security Now! episode ever. Okay? And tell your friends about this. And now of course I'm responsible for my family's cable modems, not just mine. So my mom and my sister, I've got to figure out how to fix it for them. She's on Cox, so she probably has a - and she also has Cox phone service. So it's going to take out her phone service. That's a big issue. I bet you there's FCC rules about that, taking people's phone service out?

Steve: Yeah. For example, I know that when my neighborhood has a planned power failure, there are Cox trucks with generators running on all the little substations in order to keep the phones up.

Leo: Terrible idea. My mom said, "You told me to go do that." I said, "No, I didn't. I said the exact opposite. Do not use your cable company for your phone service."

Steve: Yup.

Leo: Okay.

Steve: I just abandoned my landlines, as a matter of fact, in the last few days. All they were was just generating constant telemarketing calls.

Leo: Right, who needs them?

Steve: And I thought, okay. I mean, and I wanted to send a message, not that anyone cares. But it's like, look, folks, you could have fixed this. You chose not to. I mean, I like a good solid wired copper connection, but sorry. This is just ridiculous. I don't even answer it anymore.

Leo: Well, we'll keep our eye on this one, and you keep listening to Security Now! for this kind of very valuable information. We do Security Now! on Tuesdays, 1:30 Pacific, 4:30 Eastern. That's 21:30 UTC, part of a long day. So sometimes it gets pushed, often it gets pushed back a little bit. But just, you know, watch all day. You know? Just watch all day. You can see or listen to our live stream at TWiT.tv/live.

There's also, of course, Steve's site, GRC.com, where you can get SpinRite, the world's best hard drive recovery and maintenance utility. Steve, you haven't been plugging it lately, but I'm going to plug it for you. Everybody needs a copy of SpinRite, if you've got a hard drive. Especially these new - I now have a 16TB hard drive, two of them.

Steve: I heard you say that. It's like, okay.

Leo: 450 bucks, 16TB, Steve. It's amazing. But that's a drive you really need SpinRite on before you use it.

Steve: Well, the new SpinRite. The new SpinRite. You can't even start on the old SpinRite now.

Leo: Well, hurry up and finish SpinRite. I got a bunch of those.

Steve: Yup, I'm on it.

Leo: Can I do an 8TB drive on it?

Steve: Well, I mean, you can run it, but it'll take...

Leo: Forever.

Steve: The problem is it's - yeah. Now, the new one runs half a terabyte per hour. But that was actually the lower density drives. Essentially the new SpinRite, the forthcoming SpinRite, will run at the drive speed. That is, the maximum speed the drive can go.

Leo: Nice. Nice.

Steve: So the way we have gotten 16TB is the density has gone up so high.

Leo: Yeah.

Steve: So the density goes up.

Leo: So, and those helium drives, they don't even have air inside them. I mean, it's crazy.

Steve: Yeah, yeah, yeah, yeah. So anyway, that means that the throughput should go up. So we may be able to do a terabyte an hour. And so it'd be feasible to run SpinRite before you put it in. Even occasionally.

Leo: My Synology, when you add the 16TB drive, does a parity check all the way through, and it took it days.

Steve: Oh, my god, as it would.

Leo: Now I want to put another one in, but I don't know. Oh, what a world. We live in an interesting world, and this is the place you can find out more about it. GRC.com. He has 16Kb audio, which is great for bandwidth impaired. He has actually the smallest version of the show, which is the human written transcript. Elaine Farris does a great transcript. That's a quick download. A lot of people like to read along while they listen, and then use it for later study, underline it, annotate it, put it in a book, put it on the shelf. That's all at GRC.com, along with SpinRite and a lot of other great free stuff, including what was that address for the cookie checker? Grc.sc, which is his shortcut, grc.sc/cookies.

Steve: Cookies, plural.

Leo: Cookies plural, okay.

Steve: Cookies.

Leo: A lot of great - and then /windows10? Win10, lowercase.

Steve: Yes.

Leo: There's a lot of great stuff. Steve's just - he's full of it. Great stuff, that is.

Steve: I'm full of it. All right.

Leo: GRC.com. We have copies of the show, 64Kb audio. We've got video, too, at TWiT.tv/sn. It's also on YouTube. Best thing to do, get a podcast application. Subscribe. That way you'll get it the minute it's available each and every week. You do not want to miss an episode of this show. It's kind of "must" listening for anybody who's concerned about security online. Steve, go enjoy the second half of your "Making of Star Wars" documentary.

Steve: Yay. I'm going to, yes.

Leo: And I will see you next week, right here.

Steve: Okay, buddy. Thanks. Bye.