SNI Encryption

Description: This week we look at additional changes coming from Google's Chromium team, another powerful instance of newer cross-platform malware, the publication of a zero-day exploit after Microsoft missed its deadline, the return of Sabri Haddouche with browser crash attacks, the reasoning behind Matthew Green's decision to abandon Chrome after a change in release 69 - and an "Ungoogled Chromium" alternative that Matthew might approve of - Western Digital's pathetic response to a very serious vulnerability, a cool device exploit collection website, a question about the future of the Internet, a sobering example of the aftermarket in unwiped hard drives, Mirai Botnet creators working with and helping the FBI, another fine levied against Equifax, and a look at Cloudflare's quick move to encrypt a remaining piece of web metadata.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-682.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-682-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. I'm back. And we're going to talk about a lot of things, including some things about the new Chrome that are making people upset, the bifurcation of the Internet, and a hack that takes advantage of a massive flaw in a NAS from Western Digital. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 682, recorded Tuesday, September 25th, 2018: SNI Encryption.

It's time for Security Now!, the show where we cover your security and privacy online. I'm Leo Laporte. Yes, I'm back. He's never left, Steve Gibson. Do you ever go on a vacation ever? I don't remember you ever going on vacation.

Steve Gibson: I never have, no. There's nowhere I would rather be than right here, working on code and...

Leo: Oh, come on. Come on.

Steve: No, it's true. It's sad. It's sad, but true.

Leo: Well, it's not sad. It's actually great. It's certainly more economical.

Steve: Well, I'm just happy where I am. Lorrie's figured out...

Leo: But, yeah, doesn't your - yeah. Doesn't Lorrie want to go somewhere?

Steve: She would love to, and she's figured out, well, okay, I'll just find some girlfriends and go with them.

Leo: Go by herself, yeah.

Steve: I said, "Great, honey. We'll FaceTime."

Leo: Where does Lorrie want to go?

Steve: Everywhere. Anywhere.

Leo: See, I love to travel.

Steve: She's normal. She's normal.

Leo: Where should we send Steve? You should go to Russia. You know where you should go is Estonia, the digital capital of Central Europe.

Steve: They're really ahead of everybody else.

Leo: Yeah, yeah. You'd be welcome there. People would gather 'round. Hasn't anybody ever invited you to a conference or anything like that? Of course they have.

Steve: No, I get that all the time, yeah.

Leo: And you just don't go.

Steve: Yeah, it just doesn't make any sense. It's like, you know, I think the last thing I went to is RSA when I discovered Stina and Yubico. So that was good. But anyway...

Leo: This is the guy who's lived on ham sandwiches for 20 years.

Steve: We're at Episode 682, the last episode of September. And Leo, you missed it last week because the Presidential Alert was originally scheduled for last week, but it's been postponed until next Wednesday.

Leo: But wait a minute. What do you mean? What Presidential Alert?

Steve: We talked about this on the TWiT that I did with you two years ago.

Leo: Well, I know the capability exists.

Steve: It's happening. Trump is going to tweet.

Leo: Why?

Steve: It's actually, well...

Leo: Is it a test?

Steve: Well, it's a FEMA test. And every cell phone which is turned on. And apparently the title is Presidential Alert. And it starts out...

Leo: It's the one alert you can't turn off on any phone.

Steve: Correct, correct.

Leo: Supposedly only to be used in grave, dire national emergencies.

Steve: Well, but of course you've got to push the test button every so often.

Leo: We've never - has it ever been tested? Never.

Steve: Never been tested. It's the first time it's ever going to be done. And it was the weather back in the Southeast caused FEMA to say, you know, we don't want to confuse people with a test alert during a crisis. So they're bumping it to October 3rd, which is Wednesday, a week from tomorrow.

Leo: Is it during - what time is it?

Steve: Oh, it will be. Yes, it'll be during your Wednesday podcast. I think it's some bizarre time like 2:18 p.m. Eastern.

Leo: Oh, great. It'll be right in the middle of Windows Weekly. Oh, boy. Well, that's good. Now, there's extreme, extreme penalties for rebroadcasting an EANS alert on the radio. And I know this because some DJ in New Orleans did it on - he worked for the company, the radio company I work for. And as a result, every three months I

have to take a test, which mostly consists of "Is it ever okay to rebroadcast a sound that sounds like the EANS signal?" And the answer is no. "Thank you, you passed the test, and then we'll talk to you again in six months." But this is going to be coming over the phone. I presume I could just hold it up to the microphone; right?

Steve: I don't know if it makes any sound. It's a text message. So, you know, it's an SMS.

Leo: Oh, no, it makes a sound. Oh, you haven't - you must have turned off the Amber alerts.

Steve: Oh, yeah.

Leo: It makes a loud, annoying sound, particularly if every cell phone in the room goes off at the same time. It's extremely loud.

Steve: Either we don't get them, or everybody I know has turned them off because I've never had that...

Leo: Yeah, everybody's turned them off. But you can't turn this one off.

Steve: Well, there's an effective alert system.

Leo: You will hear a loud noise. It will go [mimicking loud buzzer]. And then it doesn't say anything, it just pops up.

Steve: It's like our cable TV provider. It's like they're saying it's a monthly test, but I swear it's like every other week.

Leo: Oh, that test I hate.

Steve: It just shuts everything down and [mimicking loud buzzer]. It's, like, so annoying.

Leo: So just out of curiosity, does the President get to write the announcement? Or is FEMA going to do it?

Steve: No, no. This one, this test, is prewritten. We know what it's going to say. I had it in our show notes because...

Leo: Because you know if President Trump was given the right to write that, he would do something wild.

Steve: Yeah, he would have fun with it.

Leo: So to speak.

Steve: Yeah. And it was almost three years ago that you and I and - who was our co-host? Because I came up for the holiday TWiT.

Leo: Yes, right.

Steve: Or tweet or whatever. TWiT, yeah. And that was when we got the news that this was a new thing that was going to happen. And of course we'd had the election already, so we knew that it was going to be President Trump who would then have his finger on the button.

Leo: Yes, we talked about it, that's right. That's right.

Steve: Yeah, exactly. So anyway, the first one happens a week from Wednesday.

Leo: Well, that'll be interesting.

Steve: In the meantime - yes, it will. And I'll remind everybody next week, of course, too.

Leo: Yeah, please do because I have to be prepared for it on Wednesday.

Steve: Yes, you do. So we've got - you missed a little bit of news. I realize you were sort of offline probably.

Leo: I did not follow what was going on at all.

Steve: One of the things that Google did that was quite controversial...

Leo: Oh, yeah, I did follow this.

Steve: ...was they decided to stop displaying the www dot in front of domains that actually have that. And they also decided - they called them "trivial domains," that one and "m dot." Well, there was a huge outcry because they just sort of unilaterally surprised the world with Chrome 69 that sort of just stopped showing it.

Leo: Well, Safari's done that for a long time, but you can turn that feature off. Safari doesn't even show - they don't even show the domain name at all. And I remember I

interviewed Tim Berners-Lee, who created this whole mess. And he said, "I never expected any of this stuff to be visible to the end user. This was for computers."

Steve: Yes. And we've often talked about how hostile https://www dot, it's just like, our moms should not have to be dealing with that. So we've got additional changes coming from Google's Chromium team that we're going to talk about. We have another powerful instance of newer cross-platform malware. Now it's all being written in Python so they can have multiplatform binaries in order to have a longer reach. We've got the publication of a zero-day exploit after Microsoft missed its deadline a couple weeks ago for September's Patch Tuesday. We're all expecting maybe they're going to make it for October. We've got the return of Sabri Haddouche, who you also wouldn't have heard about, probably, Leo, while you were on vacation.

Leo: No.

Steve: But he's the guy who was, while looking for DDoS attacks on web browsers, figured out how to crash iOS. And I tried it last week, both pre- and post- version 12. And sure enough, it crashed Safari and took iOS down with it. So he's back with some DDoS browser attacks, which is what our Picture of the Week refers to. Also, we have the reasoning behind Matthew Green's decision to abandon Chrome after a change that they also made in release 69 which has also let loose a firestorm on the 'Net.

Leo: Yeah. This is the one I thought you were talking about, yeah.

Steve: Yeah. We've also got something called the "Ungoogled Chromium," which is an alternative to Chrome that Matthew might approve of. We have Western Digital's pathetic response to a very serious vulnerability, a cool device exploit collection website that I just discovered when I was pursuing that previous story. A question about the possible fracturing of the Internet in the future that Eric Schmidt brought up at a conference last week. A sobering example of the aftermarket in unwiped hard drives. The Mirai botnet creators have been hired by the FBI. Another fine has been leveled at Equifax.

And believe it or not, if we have any time, we'll actually get around to the topic of this week, which is SNI Encryption. That's the Server Name Identification which is still being sent in the clear, even with the latest TLS v1.3 encryption, which is sort of metadata. You know, we've often talked about how, yeah, stuff is encrypted, but you can still see who you're sending it to. And so although you can't see into it, you can still see like the envelope. So Cloudflare has quickly moved to adopt an interesting addition to TLS v1.3. And Cloudflare just continues to do really neat stuff. So I think another jam-packed episode of Security Now!, with lots of interesting stuff for our listeners.

Leo: You've got us all champing at the bit. I can't wait to hear all of that stuff. Before we do, though, have you installed iOS 12? I guess you have, gather you have.

Steve: Oh, yeah, yeah, everywhere, yup. And I agree with you...

Leo: And, now, I remember you hated 11.

Steve: Well, what I hated was that 10, the last version of 10 where they crippled the performance, preemptively fearing that the battery wasn't strong enough.

Leo: Right.

Steve: And on that 6S or that 6 Plus or whatever it is, the big boat that I have, even now, now we have that beta of the battery test, and it says 100%. Your battery's in perfect condition. Even so, they forced me to go to an iPhone 10. However, I'm quite happy where I am now. But yeah, so iOS 11 kind of began to get it better, and it runs just fine on 12. And I'm with you and Megan. I really like 12.

Leo: I do, too.

Steve: I like the things that they've done.

Leo: I keep getting calls, Steve, from Apple. I don't know if you heard me mention this on Windows Weekly yesterday.

Steve: I did, I did.

Leo: Got this this morning. I could play it back for you. It's the funniest thing. I just got another one while I was doing that ad. It basically says this is Olivia calling from Apple Support. Your iCloud's been breached. Don't, whatever you do, log into your iCloud. Don't use your Apple device. Call us at this toll-free number. Which keeps changing. So I just thought, well, I guess it could be true. So I logged into my - because I figured, well, if it is true, I'll certainly have, you know, I'm using two-factor. I doubt anybody's compromised my iCloud. But if it is true, I'll certainly have an email or something from Apple. So I log onto iCloud, and of course everything's fine. It's just a scam.

Steve: Yeah. I wonder if they've, like, used up all the Microsoft support people.

Leo: I think that's probably it; right?

Steve: Because they've switched over to Apple.

Leo: Unbelievable. Unbelievable. And they've called me now five times.

Steve: And it is interesting that, I mean, it does plant that seed of doubt.

Leo: Yes. Yes.

Steve: It's like, well, I don't think this is real, but, you know?

Leo: It's scary, yeah.

Steve: And I get spoofy things sometimes, too. And you do just, like, your Google account has been compromised. And it drives a savvy person to go just make sure that isn't true. So at least we're not clicking on the link and just saying, oh, yeah. Give me a call.

Leo: And the reason I bring it up is I know anybody who listens to this show is going to go, yeah, right. But tell your friends and family they're doing it because Apple just announced new products. Everybody's getting new iPhones. And so there's some credibility to it. So tell your less savvy friends and family don't believe this thing. It's bogus. It's not real.

Steve: Yeah. So our Picture of the Week relates to a story that we'll be getting to in a minute. When we talked about the iOS crash last week, Sabri had not put all of this together. Now there is a site where you, too, can go to crash your browser. It's ReaperBugs.com, R-E-A-P-E-R-B-U-G-S dotcom. And if you go to - it doesn't crash the moment you get there. You get to choose which browser you want it to stress. And so there's Reap Chrome, Reap Safari, and Reap Firefox.

Leo: I love it.

Steve: And we will get to this in more detail later on. But it is just sort of a - it's sort of a kick that he created a website to host his various browser crashes. And it probably isn't such a big problem. I mean, again, we'll talk about that. But for the title on the picture I said, "First we crash; then we burrow inside." And we do know that there are some classes of crashes which can be the beginnings of the development of an exploit. But in a world where our browsers are running code with ever-increasing performance, it's not necessarily the case that we're doing anything other than just abusing the power that code from some specific source, typically untrusted, is providing.

Leo: Is it safe to do this? Should I, I mean, this is a live picture from my Safari. Should I...

Steve: Well, if you don't mind crashing it, it does crash it. I mean, it'll just - and we know why. It's a WebKit problem where one of the newer features of CSS allows the blending of the background. And the guy created some, I don't know how many hundreds of nested divs, each which invokes this blend operation and brings iOS, or brings Safari and in turn iOS to its knees.

Leo: Oh, I clicked the wrong button. I'm in Chrome.

Steve: Ah.

Leo: That's why nothing happened. Let me refresh and do it in Chrome. There we go. Okay. Are you sure, he says. Yes, crash me. Now, I am on Mojave now. I don't know if that will protect me.

Steve: Ah, that's interesting. And they just updated that. Because I did check it, and it crashed 11, whatever the last version of 11 was. And then also I updated to 12, and it crashed that, too.

Leo: Well, it looks like Google might have put in some protections because it doesn't seem to be crashing anything. So, huh.

Steve: Okay. So speaking of Google, they're continuing to mess with Chrome's UI. So first they took away www and "m" as the prefixes. Then they came back. Then they announced that they were going to still take away www, but "m" is remaining. What they learned kind of the hard way, and this is why many people who were really upset with this a couple weeks ago argued that this should not just be something that Google can do unilaterally. You know, messing with the address is messing with Internet protocol. And, for example...

Leo: They're not messing with the address. They're just messing with what the Chrome displays; right? I mean, the...

Steve: Well, except, for example, the "m" signifies you want the mobile version of a site. So the user has some control over whether they want "m dot" or not. So Google sort of missed this. They were just assuming that www was superfluous. And they learned that, whereas www may be, "m" can actually have some important meaning. And so, yeah, I mean, how many times have we talked about that this is not the way this should have all evolved, where we've got all this cruft on the beginning of our domain names. And I talked about how GRC used to accept either URL and always took you to the same place.

But then we ended up with Google search results being a mixture of www and non-www. And so, like, links were all diluted and confused. And so I began redirecting everyone to www. Now I'm sort of regretting that decision. It was maybe 15 years ago, and I maybe should have just said, okay, if you put in www I'm going to redirect you just to GRC.com. But anyway, Google has now decided that the file:// is also - that doesn't make any sense to put in the URL.

So with Chrome 70, the next release of Chrome, they're going to be adding what they call a "chip," which is the visual flag to the left of the URL, which will just say "file." And then they're going to reformat it so it just says, for example, c:\, you know, the normal path that you see in your OS. So that's another thing that they're going to be changing with Chrome 70.

Leo: But to be clear, this is merely what is displayed to the end user; right? I mean, there's still internally file://. That's the protocol. And there's still internally "m dot" whatever, or www dot whatever; right? It's just what's showing up in that browser URL bar; right?

Steve: Correct. But the idea was...

Leo: It's an indicator, I understand, to the end user that this is what you're getting.

Steve: Right. Except if you do remove something like the "m dot," which is semantically significant, that is, users may need to know are they at the mobile version of the site or not. So is it prefixed by "m dot"? And so in hiding this, Google was preventing users from seeing something relevant to where they were.

Leo: Yeah. You know why they don't, because it's not a standard. It's a practice.

Steve: Right.

Leo: And some people write "mobile." Some people write "m." There's no standard on what that prefix should be; right?

Steve: Well, except that so the purists, you know, the Richard Stallmans and so forth of the Internet, I mean, really got upset with the idea that Google was deciding that www dot was so-called a "trivial subdomain" and could simply be removed from the URL. I mean, the rest is still being displayed, but they just decided, well, this is not - it's just superfluous. And so as a consequence of the outcry, it came back. And so but they're still saying that they're going to talk to standards bodies and see about coming up with some sort of uniform representation that everyone agrees upon. I mean...

Leo: Apple did this a long time ago. They stopped showing the fully qualified domain name in the browser bar. They just showed you, if you're at Yahoo, you're at Yahoo. It doesn't say Yahoo.com, even.

Steve: Right.

Leo: And then there's a setting. You can turn that off.

Steve: Yes.

Leo: But I think Apple's attitude was, well, users don't need to see all that extra cruft. Especially since it's not a standard.

Steve: Right. And I think where we are is just sort of, you know, in sort of this awkward place where we're moving into a different place where the standards that we've had that built the Internet are being, like, questioned, exactly like this. Is there a need for www on the front of the URL? And at this point we don't have a consensus. And so people got upset because Google just decided, well, we're going to change this. And they are the major browser on the Internet. They are the majority browser. And what was I going to say?

Leo: While you're thinking, let me just show you what Apple has decided in this regard. So I have it checked right now, "Show full website address." But I can

uncheck that. And then instead of seeing up here the fully qualified address, I'm just going to see Twitter.

Steve: Yes.

Leo: Or Twitter.com. If I turn that on, I'll see this long URL. And you know what's missing is the actual URL for the site. It's just showing you're on Twitter. You can, if you copy it, you will get the fully qualified URL and everything after it.

Steve: Yeah. Well, and you know, Leo, I think partly this is also a reaction to how incredibly ugly URLs have become.

Leo: Yeah, yeah. Look at the difference. Do you really want to see this? But you could make the argument, I could make the argument, well, yeah but I need to know this is the Verge's account, and this is a particular status from the Verge, instead of just, oh, generic, bland, I'm on Twitter.com. At least Apple gives you the choice here. But I don't think...

Steve: And so giving the user choice, yes.

Leo: But the default is, by the way, the simplified version. And I bet you 99% of Mac users don't even know that. They don't. They're not aware of it.

Steve: Right. So what's interesting is that apparently Lawrence Abrams of Bleeping Computer picked up on this. Apparently Google is experimenting with the same thing on their search results. In his posting, Lawrence wrote: "Google really wants to get rid of the www subdomain." He says: "First we had Google removing www in Chrome 69 address bar, and now there is some test underway to remove it from search results, as well." He wrote: "I was first alerted to this when one of BleepingComputer's reporters noticed that the BleepingComputer domain was showing up in Google search results as just https://bleepingcomputer.com," although officially it is www.

He said: "When I checked from my end, though, it was showing it listed as normal with www.bleepingcomputer. While researching this behavior, I found many domains where Google was removing the www subdomain in the search results. Once I performed a refresh of the page" - meaning the search results page though - "the normal www subdomain would be listed again. In some cases, I could refresh over and over," he wrote, "and the results would switch back and forth between www and non-www.

"Ultimately," he says, "I could not get BleepingComputer.com to show the non-www version, so I found another site that was also performing this behavior. When I searched for Palo Alto Networks, it showed the domain listing without the www subdomain. If you clicked on the search result, the site would perform" - and this is interesting to me - "a 301 redirect to www," which means they weren't just changing the visual, they were choosing not to - the URL itself did not have the www in it, so that Palo Alto Networks was doing a 301 redirect over to www. He says: "...which is the site's desired behavior."

He said: "On a refresh of the search result page, the normal www version of the URL appeared again in the search results. This time, though, the site links have been changed," he said, "to a smaller display under the domain description." And he goes on.

But my guess is, as we know, Google has a massive server farm, and he was probably getting different servers picking up his query when he refreshed; and some of them for whatever reason were not showing www, and others were. And he did ask Google. He got no response from Google one way or the other to what they were doing and what was going on. But we do look like we're in some sort of flux here with whether or not or how Google is going to treat the www prefix on domain names.

Leo: It's an interesting question because - and I would hope, and this isn't the time to do it, but maybe at some point we could talk about how this all works. Because I don't need - www used to be the thing before the domain name.

Steve: Right.

Leo: Mail.www.whatever was the name of an actual machine; right? Is that...

Steve: Correct. Yes.

Leo: I think for a long time that's not been the case. And in most cases a site doesn't need www dot anything.

Steve: That is correct, yes.

Leo: That was to indicate that you're going to go to the machine that has the web server Apache's running on that machine. If you want to go to my email server, Postfix is running on mail dot.

Steve: Correct.

Leo: But nowadays that's all handled with DNS in an invisible way; is it not?

Steve: Well, actually it's more likely handled by port numbers because we have well-known ports. And so, for example, GRC's news server is at news.grc.com. It doesn't need to be because the news server uses port 119. And so it could just be GRC.com, and the news client knows to connect to port 119, in the same way that the web client knows to connect to 443 or 60. So you're right, Leo. It's sort of an old-school, you know, you have the domain name, and then you have the machines that are accessible on that domain name. And there are supposed to be prefixes of the domain name. But it's not necessary.

Leo: No. It's not handled that way anymore.

Steve: And in fact I can't even think of, I mean, now, okay, the "m dot," there's a place where, as you said, by convention rather than by standard, by convention the "m dot" has been a prefix which takes you to the mobile version of a website. But even there it could be the same IP.

Leo: Right.

Steve: But when the URL comes up with an "m dot," it gives you different content, which is optimized for mobile screen sizes.

Leo: Well, the perfect example is Twitter. If you go to m.twitter.com, you get a progressive web app for Twitter which is intended for mobile. But you know what, I just did it, and Twitter rewrites it as mobile.twitter.com. So it's not in fact the canonical name of that server.

Steve: Right.

Leo: And it's rewriting the name anyway. And there is no standard for what "m dot" means. So I think Google's attitude was, well, why should the end user be concerned about this? My only concern - I think it's probably yours, too - is from a security standpoint you kind of do want to know the fully qualified name that you're going to or that you're at or that that link points to. Right?

Steve: Yeah. I would say you at least want to be able to get to it if you want it. When Google was doing this - and I did experience it briefly. It was funny, too, because, Leo, I put in www.GRC.com, and sure enough it showed me GRC.com. Then I put in blog.google.com, and it converted it to blog.google.

Leo: Right.

Steve: And I thought, wait, what?

Leo: They own dot google.

Steve: I thought - exactly. First I thought they were taking away the dot com. And so then I put in blog.grc.com, and it didn't do anything. So I thought, wait a minute, what's going on? But it then, as you and I had talked about once before, they have dot google as their top-level domain. So, yeah. But when it was doing that, I could click on the URL twice. I think once did something, but it didn't show it to me. And when I clicked again, then it showed me the unredacted full URL. And it was like, okay. So, I mean, so it was available if you wanted to have it. And even then there was a setting deep down in the Chrome://flags that allowed you to get to that and turn that behavior off. So purists could still have it their way.

Speaking of Palo Alto Networks, which was the domain that Lawrence at Bleeping Computer thought to try next, they've found this malware that I was referring to. They named it XBash. It's written in Python. It's like the Swiss Army knife, unfortunately. It's a self-propagating worm which targets both Windows and Linux systems. It's a botnet. It's a cryptocurrency miner. It's a ransomware spoof that deletes a victim's databases and demands payment while being unable to restore the destroyed data, thus the spoof of ransomware.

It was found in the wild by the guys at Palo Alto Networks, and it's been tracked back to a Chinese-speaking APT, an Advanced Persistent Threat actor known for previous cyberattacks using similar attack mechanisms. They said it has nascent functionality, without specifying what, that would allow it to spread quickly within an organization's network, but they hadn't - but that was, like, still not activated in the code that they had seen.

What's a little disturbing about this, I mean, you know, you don't want any of this. Nobody wants anything like this on their network, on their servers. But it hunts for vulnerable, unprotected web services using well-known ports and deletes the data from many different popular databases, including MySQL, the Maria database, Couch, and Mongo running on Linux servers. It scans a targeted IP, both over TCP and UDP, for the well-known ports HTTP, VNC, the well-known database ports, Telnet, FTP, RDP, ElasticSearch, Rlogin, and others.

So it's, like, scouring for any way in. If it finds a listening service, it uses a brute-forcing, assuming the use of weak usernames and passwords in a dictionary attack, and apparently it contains a large dictionary, which it's finding effective in getting in. And then, if it is in, it deletes the databases that it's able to gain access to, completely just deletes them. Doesn't encrypt them, like misbehaving ransomware. I was going to say "well-behaving ransomware," but it's misbehaving. But in this case it just deletes them and displays a ransom note asking for payment, even though it does not have the capability of restoring it.

Leo: It's deleted.

Steve: Like, why bother with that functionality? We've already got their money. Good luck.

Leo: Yeah, of course.

Steve: So it's known to have infected at least 48 victims who have apparently, says Palo Alto Networks, it's not clear how they know this, but who have paid the ransom, totaling about $6,000, and never received their data back. There's no indication that any data recovery happened afterwards. Anyway, so the...

Leo: I paid my money.

Steve: Exactly. Can we please have our SQL database back? Oh, goodness. Okay. So it's written in Python. And then the PyInstaller system is used to convert the malware into multiplatform, standalone, executable binaries which can infect Windows, Apple macOS, and Linux. Which of course allows it to run cross-platform everywhere.

Okay. So I was thinking about this. Okay. So here's the news. Multiplatform, really bad malware - oh. On Windows it only runs a cryptocurrency miner. It does not install its botnet functionality. But it does that under Linux. And it wasn't clear whether it deletes Windows databases or only Linux databases. But still, so what's our takeaway? And I was thinking about this. First of all, we know no services should ever be publicly exposed unless they need to be publicly exposed. Like, for example, there have been so many instances where somebody's database is installed on a server. And what is the MySQL

port, 3369 or something? I can't remember now what it is. But it just - it opens the database service port on the WAN interface. It's like, it's crazy.

And, I mean, it is the default when you install MySQL from Oracle. It's like, oh, yes, we're going to make these services available. But hopefully any admin who knows what they're doing is saying, okay, we don't want to expose - we don't want to put that on the Internet. So first of all, no service that doesn't have to be publicly present should ever be.

And the other thing is, think about it, we're well past the point where users are logging in; right? Other client systems are connecting to a database backend. Other client systems, even if it's a news reader, you know, your news reader, not you, is connecting to the news server. So all of these usernames and passwords should be, as long as they can be, absolutely random gobbledygook. It's like go to GRC.com or wherever, like our passwords page, and just get some gibberish. That should be what you use if you have to have a service publicly exposed. No person ever is going to be typing that in. Some other client is going to be connecting to the service.

So the idea that usernames and passwords still need to be user memorable, like that ship as sailed. And yet, obviously, for this thing to be doing a dictionary attack and for it to be successful says that people are still setting up servers imagining that a user is going to be manually logging into something, which is almost always a backend for some other client, which then can be given the responsibility of remembering how to log in.

So just something for our listeners to think about is, you know, as you were just saying, now that we have a world where we have LastPass to do the logging in for us, I was explaining last week or the week before that to sort of put my own work with SQRL into context, we're still in a mindset of a dumb terminal where back in the day, when you and I were at college, Leo, there was some CDC mainframe, and we were using a Hazeltine or a Lear Sigler or some terminal. And we were, like, logging in with our username and password because this was a dumb terminal.

Now we're all using computers to log in. So on one hand we can have password managers that remember for us; or, in the case of SQRL, the work I'm doing, we have computation. There's some computability at our end, so it's feasible for the thing we're logging into to send us a challenge which requires some computation on our end. Which now everybody has. In every device that we're logging in with, it's not just a memory machine, it's a computer. So anyway, I just sort of want to further plant this notion in people's minds that the era in which it's necessary to remember a username, and especially a password, is really - that's just long gone. Yet obviously people are still doing that.

Leo: Not really long gone. It just should be long gone.

Steve: It's not as long gone as we wish, yes. Okay. One more, and then we'll take our second break. Microsoft missed a vulnerability deadline. So there is now in the wild, with a proof of concept posted on GitHub, a zero-day vulnerability for the Windows Jet database.

And it must be that Microsoft wasn't that worried about it because Trend Micro responsibly disclosed on May 8th to Microsoft that they had found a way in which, if the Jet database engine - which is now integral into all versions of Windows. It's down there and used for various things in Office and Access and so forth. If it opens a malformed database, there's a buffer overrun which has been found which can cause the Jet database engine, and we know where this is going, to overflow a buffer and run code

that is also contained in that database file for a remote code execution vulnerability. On the other hand, you have to somehow arrange to get the Jet database engine to do that. And hopefully, once again, it's not publicly exposed.

But so what happened was, on May 8th, Trend Micro's Zero-Day Initiative reported the vulnerability to Microsoft, and Microsoft acknowledged. Let's see. On the 14th of May, they successfully reproduced the issue and confirmed. Then nearly four months goes by. On September 9th they reported an issue with the fix, whatever that means, meaning like, oops, maybe they weren't going to have it out in time. I don't know what the issue was.

Trend Micro said, well, you know, time is up on this, and we're going to hold you to it. On September 11th they confirmed that the fix did not make it into the build for September's Patch Tuesday. And on September 12th Trend Micro confirmed their intention to release it, pursuant to their zero-day schedule, which they did on the 20th, which, what, was last Monday? So, wait, no, the 20th. Today's the 25th. So middle of last week.

So at this moment on GitHub is a full exploit proof-of-concept for a Windows zero-day which is effective across all versions of Windows. We can presume that this has Microsoft's attention. I mean, Microsoft knows about it. And it felt like they were trying to get it into the September Patch Tuesday. We can presume it'll be there in the October Patch Tuesday and also that Microsoft's not that worried about it. They're saying that user interaction is required to exploit the vulnerability inasmuch as the target must visit a malicious page or open a malicious file. On the other hand, visiting a malicious page is not a high bar to reach at this point.

So I imagine probably next week, which will still be before Patch Tuesday, we may be talking about some in-the-wild exploitation of this Windows zero-day because what we're seeing now, the trend is bad guys know they don't have a large Window of opportunity before these get closed. And so they're just jumping on them immediately. This is worse than a privilege escalation or elevation. This is a bad guy running their code on your computer. And if you can actually do that with a web page, then I think that's pretty serious.

Leo: Yeah. And you're right, I mean, many web pages that are even responsible, well-run web pages sometimes have malware on them.

Steve: Because they host malicious ads.

Leo: Yeah, right.

Steve: Because they just have URLs to some ad service; and it's like, oh, show anything you want to here.

Leo: Right, right.

Steve: So, okay. I don't know how much we care about this, but if nothing else it's a fun hack. We talked last week about Sabri Haddouche, who, while looking for browser DoS, browser Denial of Service things, he encountered this iOS hack which, when I tried it, as I said, brought my iOS, my Safari WebKit, because basically it's a WebKit vulnerability,

brought it to its knees. I don't know if it's a vulnerability. So, for example, in that case, there's a feature of CSS which allows a CSS item to ask to have its background, that is, the stuff behind it, blurred or color skewed or made into monochrome. It's fancy. So it requires image processing to display this object.

And what occurred to him is that, since you can nest divisions, "divs" as they're called in CSS or HTML, you could set the CSS, the cascading stylesheet, the style of the div element, to require blurring of its background. And if you then nested them, you would be nesting these blurrings or whatever image processing you required. And to faithfully do that, you'd have to go to the farthest back division and have it perform its blurring, and then have the resulting image given to the next higher in the hierarchy nested division and have it do its blurring.

Anyway, you could see that this becomes very processor intensive. So unfortunately it crashed. And I guess I would argue that it's bad if somebody can design something where you could just go to a website, and it crashes you out. I mean, it doesn't just freeze your browser, it reboots iOS. At least it did then. And here on this web page at ReaperBugs.com, he's got three buttons: Reap Chrome, Reap Safari, and Reap Firefox. And it looks like, on Chrome, it looks like it'll bring 69, both Chrome and Chrome OS 69.0 and earlier, which is where we are now at Chrome 69, it'll crash it. And for Safari he shows iOS and macOS from 9 to 12.0, it'll crash those. And also there's one for Firefox, 62.02 and earlier. And I don't remember where Firefox is right now in its - let's see. About Firefox, yeah, 62.0.2. That's the latest version that is in the main channel.

So as I said before, if this was a glitch in JavaScript processing, which could potentially be leveraged into a buffer overrun that could allow a bad guy to run code on your machine when you visit a web page, much like maybe that Microsoft Windows Jet database problem that zero-day we mentioned is, then that would certainly be a problem. But if it's just, I mean, we understand that we're running code now, I mean, for a long time we were promoting the idea of running NoScript, and that you would have NoScript on by default, and you would only allow scripting where necessary. And of course our listeners know that I gave that up.

Leo: I gave it up after three minutes.

Steve: Yeah, I know. It was just like, okay.

Leo: No site works.

Steve: Nothing worked anymore.

Leo: Congratulations, you've broken the Internet.

Steve: So we had to back off of that policy because scripting is just now a reality, which means when you go and browse around the Internet, you are running JavaScript of the site that you're visiting. Now, if the site abuses that privilege, you close the page. You say "Ow," and "I'm not going back there again." And so that's sort of a self-limiting effect. And we've talked about the idea of cryptomining. You go to a site, and it pins your CPU. Same thing. Ouch. And you close your tab. So it makes sense, I mean, I'm thinking this is nice work that Sabri is doing. And presumably Safari could put a - or WebKit could

put a special case in where, if it's like you're asking for a stack of this blurring effect, more than 10 deep, it's like, no. It's like prevent itself from doing that.

The problem is there are probably an infinite number of ways of abusing code. I mean, this is code. And in fact I was reading down through some of the conversation about this, and there was someone else who posted some JavaScript that just used the forward and backward ability. Like in JavaScript you're able to script moving to the previous page and moving to the page you just left. So you could certainly put that in a loop and just cause the browser to jump back and forth.

Leo: Yeah, right.

Steve: It's like, okay. How is that useful? I mean, and the point is we don't want to bog our browsers down with special casing every possible way because there's an infinite number, I mean, it's code. There's an infinite number of ways you can get up to mischief. So it would be nice if there was maybe some resource control, or maybe if the browser just closed the page that was abusing it itself so it's just like, okay, go away, bad page. I don't know. Or maybe puts up a sign that says, okay, this site is misbehaving. Who knows? So maybe there is some sort of prophylactic measure generically that could be taken to protect browsers.

So I think what Sabri is doing is great. Either he's going to succeed in driving some useful change, or he's just going to have a page that is always able to annoy people who click on the, yeah, crash my Chrome.

Leo: Please.

Steve: Yeah, it's like, okay. But the good thing that could come out of this is if remote code exploits are found. But if it's just a matter of making the browser unresponsive, it's like, well, okay. I'm not sure that's - maybe it's worth showing people. Who knows? And if the browsers can do something, sort of like all-encompassing, to say we don't like this website you're at.

Leo: Don't ever come back here.

Steve: Yeah. Would you like us to remember not to come back here again for you or something. I don't know. But it was sort of interesting, and I wanted to follow up on the work that he had done before. And just to mention that now you can also, if you're interested, you can crash your Chrome.

Leo: Hurt yourself?

Steve: And you could crash your Firefox. Okay. So Matthew Green, our professor of cryptography at Johns Hopkins University, whose work we often cite and enjoy, his blog is titled "Why I'm Done With Chrome." And I'll just say he's not alone. Now, he wrote lengthy, and we're not going to go through it all. It was 2,236 words, believe it or not. But I will share the beginning of it because it sets it for us and explains why he's upset and gives us our context.

He starts off with: "This blog is mainly reserved for cryptography, and I try to avoid filling it with random 'someone is wrong on the Internet' posts. After all, that's what Twitter is for," he says. "But from time to time something bothers me enough that I have to make an exception. Today I wanted to write specifically about Google Chrome" - and I should just mention I'm not onboard with this completely. It's like, come on, really? But okay. So "...specifically about Google Chrome, how much I've loved it in the past, and why, due to Chrome's new user-unfriendly forced login policy, I won't be using it going forward."

Okay. So we'll take a break here for a minute. It's like, what? After the release of Chrome 69, users discovered that anytime they logged into their Google account, or any Google service, they would also be automatically logged into Chrome, which is different than that, whether they wanted that or not. And I'll just mention that, too, can be overridden. The underlying feature here is something that the Chromium folks call "identity consistency" between the web browser and the cookie jar. And it's mostly the fact that, as you said, Leo, Google has the dot Google domain.

And we know how cookies work. Whenever your browser is making a query to a domain for which it has previously received a cookie which is still valid in the context that it's being used, because there's secure and a number of additional flags that could be added, and it hasn't expired and so forth, it offers the cookie up. And it is the cookie that creates this notion of continuity and logging in-ness and this abstraction of having a session, a login session. So this is sort of tied in, this identity consistency, with the web browser and cookie jar.

But they did change the behavior when they went to Chrome 69. And so the Chromium people say, when this is enabled, as it is by default, the browser manages sign-in and out of Google accounts under Mac, Windows, Linux, ChromeOS, and Android. So Matt Green and many others feel that this is a big deal since it associates a browser with a Google account, which he argues should not happen unless the user explicitly chooses to log into Chrome. Even if browsing data is not uploaded, and sync is not enabled, there's data that could be gathered simply by the authentication process alone.

And it's true that, pursuant to Google's own privacy policy, logging into the browser does involve a different set of privacy expectations. In their privacy policy, Google says, "When you sign into the Chrome browser or a Chromebook with your Google account, your personal browsing data is saved on Google's servers and synced with your account. This type of information can include browsing history, bookmarks, tabs, passwords and autofill information, other browser settings like installed extensions." And they say: "These settings are automatically loaded for you anytime you sign into Chrome on other computers and devices. To customize the specific information that you synchronize, use the Settings menu."

So the point is that logging in with Chrome to your browser is like a separate thing. And so what has Matt all bent, and a lot of other people, is that this is something they changed without mentioning it. He says: "The fact that Google has decided to sign users into their browser without their permission causes" - oh, I'm sorry. It's me in my notes. The fact that Google has decided to sign users causes Matthew to worry that Google may decide to start synchronizing user data whenever they choose. He wrote to Google: "If you didn't respect my lack of consent on the biggest user-facing privacy option in Chrome, and didn't even notify me that you had stopped respecting it, why should I trust any other consent option you give me? What stops you [Google] from changing your mind on that option in a few months, when we've all stopped paying attention?"

Leo: Yeah. The roots of this go back a little way because you may remember a couple of years ago Google unified all of their services under one Google account.

Steve: Right.

Leo: And there was a big change in the privacy policy. And this they did make a big deal about. They put an announcement on and stuff. So it used to be, when you logged into YouTube, that was different than logging into Gmail.

Steve: Right.

Leo: But now it's all the same. And for instance, you have a Google Plus account - oh, Apple's calling again. Oh. You have a Google Plus account whether you want one or not. You have, if you create a Gmail account, you have Google accounts across all their services. And all this is, is the last shoe dropping that now that includes, among all those services, that includes Chrome. And I completely understand what Matthew's saying is that the browser's a special case. But the answer is easy. Just don't use Chrome. I think, now, he says that the Google team hasn't given any reasonable reason for this. I disagree. I think it's apparent what the reason is, is this is what users expect. For instance, on the iPhone, when you log into Google once on the iPhone, that's the only time you log into Google. Every time you use something else, including Chrome, including YouTube, you're already logged in.

Steve: It knows who you are.

Leo: It knows who you are. Apple allows Google, in fact encourages Google, to do that. And I would submit that that's kind of what users expect. Well, look, I logged in. I shouldn't have to keep logging in. I completely agree with Matthew that this is kind of overreaching. But the good news is you don't have to use a Google browser.

Steve: Or, well, and there is a way to turn it off. And so I think one of the issues, probably I think if we could argue that Google made a mistake, it was in not notifying.

Leo: Right.

Steve: And so he's upset because there is a lot going on behind the scenes with Google. I mean, they're already sort of shrouded in, like, okay, well, where is our data going and what's being done with it and so forth. And so the idea that he takes objection to is that it was this just happened, and people discovered, hey, wait a minute, I'm seeing my face, my Google avatar, in the upper right-hand corner of Chrome, and I did not log into Chrome. And suddenly, whoops.

Now, the fact is, however, that in this mode synchronization is not enabled. So the Chromium people did take that into account. That is, if you explicitly log into your Chrome browser, then you're enabling sync. If you sort of autonomously get logged in, sync is not enabled. And so a number of people sort of missed that. And to me it suggests that they clearly understood that that would be going too far, if they just made it exactly the same as if you had enabled cross-browser sync.

Leo: It would also be annoying because a lot of people don't enable it because they don't want the same bookmarks on all the instances.

Steve: Yes, yes, yes.

Leo: And they don't want the same user interface. They don't want the same tabs. They don't want the same extensions. I use Chrome synchronization. I like it. But I think there are a lot of people say, well, no, but I don't want those bookmarks on this as well as that. And so it would be wrong for them to log you into their synchronization automatically.

Steve: Because then it would really mess up your own manually curated, yeah, setup.

Leo: Exactly. This is why I think they're doing this out of response to what users expect as opposed - but this is sometimes hard for people like Matthew to understand because he's not a normal user.

Steve: Yeah.

Leo: Normal users are - try to explain this to a normal user.

Steve: Right. Well, and...

Leo: [Crosstalk], well, of course I'm logged in.

Steve: They could have - yeah, right.

Leo: I wanted to be.

Steve: They could have, like, presented a "We would like to automatically log you into your browser..."

Leo: They should have, yes.

Steve: "...since we see you're logged into other Google things. Is that okay?"

Leo: Right. That would have been good, yeah.

Steve: And people would go, what? Yeah.

Leo: Yeah, yeah.

Steve: And then end of drama.

Leo: People would be maybe confused by that. And remember Google's probably stinging a little bit from a couple of years ago when they announced they were going to unify all the accounts. They got a lot of heat for that.

Steve: Yeah.

Leo: And now we've all gotten used to it. I think normal users expect the same behavior that they get on the iPhone, which is "I logged into Google. Why are you asking me to log in again for?" You know? It should all be part of the same, I mean, I don't like it that I have a Google Plus account. There's no way to not have a Google Plus account.

Steve: Yeah.

Leo: That's nuts; right?

Steve: Yeah. It's worth noting that, I mean, he was really bullish on Chrome. I mean, he's upset. But he says, "When Google launched Chrome 10 years ago, it seemed like one of those rare cases where everyone wins. In '08" - which is 10 years ago - "the browser market was dominated by Microsoft, a company with an ugly history of using browser dominance to crush their competitors. Worse, Microsoft was making noises about getting into the search business. This posed an existential threat to Google's Internet properties.

"In this setting, Chrome was a beautiful solution. Even if the browser never produced a scrap of revenue for Google, it served its purpose just by keeping the Internet open to Google's other products. As a benefit, the Internet community would receive a terrific open source browser with the best development team money could buy. This might be kind of sad for Mozilla," he writes, "who have paid a high price due to Chrome. But overall it would be a good thing for Internet standards." And of course we've often talked about how Google may be sometimes viewed as a bit heavy-handed, but they really do have the users' interest almost always at heart.

Leo: Exactly. We may disagree with what they do, but I don't think they do it out of evil intent. They do it because they think this is how users work. And I think that they're right. Although your next story is exactly what Matthew was talking about. The upshot, the result of creating Chrome is actually very positive.

Steve: Yup. Okay. So before we get that, there is a flag. Google calls it "identity consistency," that is, this thing we've just been talking about. So if you go to chrome://flags, and then into the "search flags" field, you type "account-con." That's enough. That will select out just this one, which is account consistency. And if you don't like this, you just disable it. Set it to disabled, restart your browser, and your Chrome will no longer auto login for you. So again, they did make it tweakable for those who don't want the default for whatever reason.

But as you were just saying, Leo, we now have Ungoogled Chromium. And the tagline is "Bringing back the 'Don't' in 'Don't be evil.'" And this is on GitHub. There are binary builds for a bunch of different popular OSes I'll get to in a second. But Ungoogled Chromium describes itself as it's Google Chromium sans integration with Google. It also features some changes to enhanced privacy control and transparency.

For their motivation and description, they said: "A number of features or background services communicate with Google servers despite the absence of an associated Google account or compiled in Google API keys. Furthermore, the normal build process for Chromium involves running Google's own high-level commands that invoke many scripts and utilities, some of which download and use prebuilt binaries" - meaning that they're just saying there's some lack of full transparency there, you don't know really what they are - "provided by Google. Even the final build output includes some prebuilt binaries. Fortunately, the source code is available for everything.

"Ungoogled Chromium is a set of configuration flags, patches, and custom scripts. These components altogether strive to accomplish three things: Disable or weaken offending services and features that communicate with Google or weaken privacy; two, strip binaries from the source tree and use those provided by the system or build them from source; three, disable features that inhibit control and transparency, and add or modify features that promote them. These changes are minor and do not have significant impacts on the general user experience."

And then they say that Ungoogled Chromium should not be considered a fork of Chromium. "The main reason for this is that a fork is associated with more significant deviations from the Chromium, such as branding, configuration formats, file locations, and other interface changes. Ungoogled Chromium will not modify the Chromium browser outside of the project's" - and they didn't say "limited," but I'm adding "limited" - "goals. Since these goals and requirements are not precise, unclear situations are discussed and decided on a case-by-case basis." And I've got a link to it, Ungoogled Chromium. I'm sure if you just Google it you'll find Ungoogled Chromium.

Leo: It's on GitHub.

Steve: Yes, exactly. And I have a link to their prebuilt binaries. They are - most of them are very current. Debian 9.0 has a 69.0, so that's a current development binary, and 68 in release. Portable Linux has about the same thing. There's also one for, oh, Mac is at 68. Ubuntu is at also 67 for release or 69 for development. And Windows 64-bit has a binary at 67. So almost completely current. And again, they're not providing those. Other people have built those for those platforms. So caveat emptor. But still, it means you're not stuck having to build it yourself, and you can use a version of Chrome that has those things done to it. So as you said, Leo, it's a consequence of...

Leo: I think it is, yeah.

Steve: Yes, of it being open source and Google saying, well, we're going to do this. If you don't like it, you can take what we've done and do something different.

Leo: I like that.

Steve: Which I think is neat. Okay. So Western Digital. They have a very popular NAS known as My Cloud. That's one of the more popular Network Attached Storage devices because it is turnkey. It's more user-friendly than some of the more techie NASes. And it's able to offer a cloud-based access; right? So here's an example of something you attach to your network. It's able to poke a hole through your NAT router and make itself available globally so that you're able to access your NAS when you're out roaming around on the public Internet. For that, you want really good security, obviously. Otherwise, bad guys can get into your NAS, which nobody wants. That's why there is a username and password.

So security researchers at Securify discovered an authentication bypass vulnerability actually across the Western Digital My Cloud NAS boxes that could allow, actually very trivially, unauthenticated attackers with network access, meaning Internet access in the case of this thing being out on the cloud, to escalate their privileges to admin level without needing to provide a password. This allows attackers to run commands that would normally require admin-level privileges and gain complete control over the affected NAS device, giving them the ability to view, copy, delete, and overwrite any files on the device. So all that's horrible; right?

Okay. So what Securify wrote in their disclosure was: "Whenever an admin authenticates, a server-side session is created that is bound to the user's IP." Okay. So you imagine somebody out on the Internet. They're logging into the NAS. And so this creates a session tied to their public IP. That's how sessions work. That's fine. They say: "After the session is created, it is possible to call authenticated CGI modules" - you know, executable modules - "by sending the cookie username=admin in the HTTP request." Just a cookie. But that's after you've been authenticated. So, they say, "The invoked script, the CGI, will check if a valid session is present and bound to the user's IP address." So that's fine.

However, what they discovered was it's possible for an unauthenticated or pre-authenticated attacker to create a valid session without being required to authenticate. Whoops. An authentication bypass. The network_mgr.cgi module contains a command called cgi_get_ipv6 that starts an admin session that is tied to the IP address of the user making the request when invoked with the parameter flag=1. Meaning that's all you have to give it. Subsequent invocation of commands that would normally then require admin privileges are now authorized if the attacker sets the username=admin cookie. And they show a proof of concept which is just, I mean, it's a standard post submission, an http post, to this manager, this network manager CGI, giving the cookie username=admin with a content length of 23 and just a single line, cmd=cgi_get and so forth, with a flag=1.

You send that to any Western Digital My Cloud NAS box, and you're authenticated as an administrator. And you can then do anything you want to from the Internet. And in their disclosure they say: "Next, call an endpoint" - meaning a command like cgi_get_ssh_pw - "that requires admin privileges and authenticate as admin by adding the cookie," and it works.

Okay. So in Western Digital's blog just recently, I got a kick out of this, they said: "Recently, security researcher Securify published an authentication bypass vulnerability for our My Cloud products." They said, "We are in the process of finalizing a scheduled firmware update that will resolve the reported issue. We expect to post the update on our technical support site at support.wdc.com within a few weeks."

Okay. Now, but get this. What I didn't say is that on April 9th of 2017, okay, 2017, Securify discovered the vulnerability. And the next day, on April 10th of 2017, they notified and reported this to Western Digital customer support and were ignored. So they waited until September 17th of 2018, I mean, maybe they forgot. Like, what, 17

months? No, 19 months later? On September 17th of 2018, this month, they requested a CVE designation for this. They received it the next day, on the 18th, and later that day - on the 18th, so what's that, that's last Tuesday - published the details of this. Then they got, finally, Western Digital's attention.

After this vulnerability was known to Western Digital, these guys kept their mouth shut about it for 19 months. And within a few days, actually, three days it took Western Digital to update the firmware once this went public. So the firmware is now available. So the first takeaway is, because as we know there are bots, and there are bad guys out there scanning for this - so this became public knowledge a week ago, and there's no authentication on any of these Western Digital My Cloud products. So if you have one, and if it is publicly exposed, you absolutely definitely want to update the firmware on your device posthaste.

And I'm at a bit of a loss about what to do about, I mean, what they should have done. It's a vulnerability that they did not disclose, to their credit. But on the other hand it wasn't until they did disclose it that Western Digital then patched it in three days. So maybe they should have held Western Digital's feet to the fire, given them a 90-day deadline and said, look, as other companies do, as Google does with the things it finds, fix this in three months because we're going public with it whether you do or not. Then if Western Digital blew them off, well, once they went public, three days later this would have been fixed. To me, that would have been way better than leaving this unpatched for 19 months, during which time it could have certainly been possible that other people would have found this.

These guys found it by reverse-engineering the code in the My Cloud firmware, looking for vulnerabilities, and they found a big one, complete login authentication bypass, and told Western Digital, who then did nothing about it. So I'm unimpressed with Western Digital, except I'm kind of impressed with their response once they decided, ooh, we'd better fix this. They were certainly able to immediately. So I don't think they should have been given this much time to fix something that is this important. And anybody who does have a My Cloud system should absolutely get themselves updated immediately.

Leo: On we go, Steverino.

Steve: So in researching the previous story about the NAS, there were some links that I was following. And I ran across a site I just wanted to put on our listeners' radar because I thought they would get a kick out of it. It's Exploiteers. So it's E-X-P-L-O-I-T-E-E dot R-S. So it's a clever use of the .rs TLD. And it's a site dedicated towards just sort of old-school hacking of all kinds of gadgets and IoT devices. For example, I just pulled one out because it was a Sony Blu-ray player, for example. They have there: "A bug exists in the MTK-supplied SDK which affects many Blu-ray players, including the BDP-S5100. The main binary, which controls all aspects of the player, has leftover debug instructions for the VUDU app. When the VUDU app is run, if a file exists named vudu.txt in a directory labeled vudu on a FAT-formatted flash..."

Leo: This is awesome.

Steve: Isn't it? It's so cool. "It will attempt to execute vudu/vudu.sh and deletes vudu.txt. It runs this sh as root. Using the commands below, you can spawn a root telnet shell, allowing access into the device." And then they just show you. It's a simple shell script, like five lines, that ends up running the telnet daemon. And so they explain: Put this on a file. Stick it on a FAT-formatted flash drive. Restart the device. Go to vudu on

the Sony menu, and now this thing then runs a telnet server. So it's going to be listening on port 23. You then telnet to the Blu-ray, and you're in. But, I mean, the list is just like LG, Panasonic...

Leo: Well, it's a wiki, so people are adding to it; right?

Steve: Yes, yes, yes, yes. And so, I mean, it's just like the Who's Who of - here they've got Google. There's a Belkin network [crosstalk]...

Leo: I have a lot of this old, obsolete hardware that's just sitting in a box. I can do something with it. That's nice.

Steve: Yeah, yeah, yeah. And so they have pictures and videos. There's people hooking wires onto the motherboards of things and doing stuff. So just sort of fun hacking stuff.

Leo: You remember the Pogoplug? Got an old Pogoplug? Well, look at this. You can gain root, turn it into a little mini server. Wow, that's awesome. What a great site.

Steve: Yeah, I just...

Leo: What is .rs? Is that Russia? Where is .rs?

Steve: No, Russia's .ru, so I don't know what .rs is [Serbia].

Leo: I like it. Exploiteers. It's Exploitee.rs.

Steve: Yeah, I just thought our listeners would get a kick out of playing with that. Okay. So this was weird. And I would argue against what Eric Schmidt believes. But last Wednesday, in San Francisco, at an event hosted by the venture capital firm Village Global, the economist Tyler Cowen asked about the possibility of the Internet fragmenting into different sub-Internets with different regulations and limited access between them in coming years. He said: "What's the chance, say, 10 to 15 years, we have just three to four separate Internets?" Eric Schmidt, who is of course the past CEO of Google and executive chairman of its parent company, Alphabet, predicted right there on the spot that within the next decade there will be two distinct Internets, one led by the U.S., and the other led by China.

Leo: Wow.

Steve: Yeah. Eric said, quote: "I think the most likely scenario now is not a splintering, but rather a bifurcation into a Chinese-led Internet and a non-Chinese Internet led by America. If you look at China," he says, "and I was just there, the scale of the companies that are being built, the services being built, the wealth that is being created is phenomenal. Chinese Internet is a greater percentage of the GDP of China, which is a big number, than the same percentage of the U.S., which is also a big number."

He says: "If you think of China as, like, 'Oh, yeah, they're good with the Internet,' you're missing the point. Globalization means that they get to play, too." He says: "I think you're going to see fantastic leadership in products and services from China. There's a real danger that along with those products and services comes a different leadership regime from government, with censorship, controls, et cetera." He says: "Look at the way BRI" - that's the Belt and Road Initiative. He said: "...the way BRI works, which involves 60-some countries. It's perfectly possible those countries will begin to take on the infrastructure that China has with some loss of freedom."

And I thought, okay, what? So I'm skeptical. I mean, first of all, we've talked a lot about the Great Wall, the firewall. We talked recently about the project that Google was outed as being working on in secret, which is to come back to China with a censored search in order to operate within the Chinese boundaries.

But, you know, in thinking about this, it occurred to me that we have an existing, much older, and very mature model in the global telephone system. And as far as I know, someone in China is able to phone someone in the U.S. and vice versa. I mean, it's not like there's a separate phone system, and they're not interconnected, and there's no way to talk across some artificial boundary. So I don't know, Leo. To me it's hard to imagine that economically you wouldn't lose so much by chopping the Internet into two pieces. It seems hard for me to imagine that that's realistic.

Leo: That's always been the argument against it, is you don't want to participate in the global economy? Okay, fine. But China's so big, in a way it is its own global economy. And remember that the phone system works that way because there are gateways. Similarly on the cell network there's a gateway between Verizon and T-Mobile.

Steve: Yeah. Yeah.

Leo: So it's not inconceivable that there would be gateways. What China's interest is, of course, is in controlling the information flow from the West. And so I'm not sure that Eric is wrong. It's not the first time I've heard this idea. It's certainly what China would like to do. Whether China will succeed is another matter because, you're right, there's all this pressure to be global; right?

Steve: Yeah. Yeah.

Leo: I don't know if it's a bad thing or not. We always had the Iron Curtain, remember? We're old enough to remember the Iron Curtain that separated the East from the West. That was, in effect, an economic block.

Steve: Yeah. So something happened recently that caught a lot of attention. PC Mag had the headline: "Unwiped Servers With Data on Millions Sold on Craigslist." Another headline: "Database servers sold at NCIX auction, allegedly without being wiped." Bleeping Computer's headline: "Unwiped Drives and Servers from NCIX Retailer for Sale on Craigslist." And so forth. Sophos, their Naked Security blog, had the story and, I thought, just very succinct coverage. They said: "Bankrupt NCIX customer data resold on Craigslist." Sophos said: "For Canadian or U.S. customers of NCIX during the past 15 years, they should assume any personal data or credit card information logged with them

is now potentially in the hands of cybercriminals and raise any suspicious transactions with their bank."

Now, for those who are not aware, NCIX was a major Canadian online and brick-and-mortar retailer. Sophos said: "What happens to sensitive customer data when a large company that has collected it over many years suddenly goes bust?" And that's where we're going to get to here in a second. But they said: "It's easy to assume that databases are wiped by diligent IT staff just before they turn off the lights and close the door for the last time."

Leo: Oh, sure they are.

Steve: Uh-uh.

Leo: Absolutely going to do that before they walk out. You bet.

Steve: See ya. And they say: "At the very least that data should have been encrypted." It wasn't. They say: "It has now emerged that something entirely different and more troubling took place when Canadian computer and electronics retailer" - and I sort of think of them as like the Fry's of Canada. They sort of had that profile. "Electronics retailer Netlink Computer Inc. declared bankruptcy in December of 2017." So, okay, at the very end of last year.

"According to Privacy Fly researcher Travis Doering, the company simply abandoned" - not surprisingly - "much of its equipment in a hurry, which he discovered when it was offered for sale on Craigslist this August," so month before last. "After arranging a meeting with the seller to examine the hardware, it turned out to comprise 20 Dell PowerEdge and Supermicro servers, 300 desktop PCs, 109 hard drives, and another 400-500 drives that had been inside those desktops or sent to it for repair.

"Now for the disturbing bit. It soon became clear that the valuable part of the data was not the drives themselves, but what was on them: 13TB of data all told, including 385,000 database records containing names, email addresses, phone numbers, and account passwords, 258,000 of which included full credit card payment details. A separate Canadian database contained 3.8 million customer records gathered by NCIX between January 2007 and July 2010. The seller had got hold of passwords to access the databases, while significant amounts of the data were not encrypted in the first place." But get this. The seller understood the value of the data and was offering it, itself, for $15,000.

Leo: They knew.

Steve: Uh-huh.

Leo: There's good stuff on here, man.

Steve: You can have all the data. Who could possibly take advantage of that?

Leo: Wow.

Steve: Anyway, so this was a bankruptcy. The landlord foreclosed, got the equipment, and so there we are. So of course this raises a good point. The Internet is driving a consolidation of retailers. And here's a big company that had a ton of data on its servers, and the question is whose responsibility is this when the company decides to walk away. It seems to me like it's the kind of thing that bankruptcy proceedings should evolve to consider is what about the data contained on your hardware? Because of course this stuff has to go through the courts in order to get liquidated. And apparently that's still something that just sort of is not being considered and is just slipping under the radar.

Leo: Do you remember - this was 2003. A guy named Simson Garfinkel and Abhi Shelat, they were graduate students at MIT. They bought on eBay 159 hard drives for $5 to $30. And this is 2003. So, you know, this is...

Steve: Fifteen years ago.

Leo: Yeah. So of these drives - one of them came out of an ATM, by the way, completely unerased. It was a year's worth of transactions with account numbers. Another one had been formatted, but of course they used simple unerase utilities.

Steve: Yeah, unformat.

Leo: Yeah, and they got 5,000 credit card numbers. Of the 150-some hard drives they bought, all but a few had - everything was still on it.

Steve: Yup.

Leo: In many cases they just deleted what was in the My Documents folder. Isn't that funny? So this was 15 years ago. It still happens.

Steve: Yeah.

Leo: Nobody's learned.

Steve: No change.

Leo: No change. We'll be telling this story in 15 years. Episode 999 we will have a similar story. The only thing that's changed is the hard drive capacity. I mean, these guys weren't that big.

Steve: Yeah. So the U.K. regulator has fined Equifax 500,000 pounds, which is in U.S. dollars at the moment $658,419.

Leo: Holy cow.

Steve: Yes.

Leo: Wow. Sharp slap on the wrist.

Steve: Well, and it's reputation, it's like further reputation damage. As we all know, and as we have often talked about because Apache Struts stays in the news because it's continuing to have some problems, although less severe than the one that bit Equifax, they suffered a significant data breach in 2017 after leaving a widely known and long since patched flaw in Apache Struts present on their servers, which were facing the Internet, which allowed bad guys to get in and exfiltrate basically all of the Equifax sensitive data. And it's always easy to pick on someone after the fact. But a company such as Equifax, which is collecting and vacuuming up sensitive financial consumer information without our knowledge and permission, I mean, I never told Equifax I wanted them to do this. They're selling...

Leo: No, but it's a contingency on getting a loan or renting an apartment.

Steve: Yes.

Leo: In the fine print down there it says we're going to provide this information to Equifax.

Steve: Right. Oh, back to the credit reporting firm.

Leo: Yeah.

Steve: Ah, okay.

Leo: Part of the deal.

Steve: Anyway, so of course it's not unreasonable for us to expect them to be really, really careful with all this data that they have collected.

Leo: Oh, of course they are.

Steve: And, I mean, this fine from the U.K. is like the least of their problems because of course they've been slapped with multiple class-action lawsuits. And, I mean, this is bad for them.

Leo: Yeah. But as of yet, I don't think Equifax has suffered any consequences. They've made more money than they've lost in this.

Steve: Yes, well...

Leo: There is a story, though, that is very good news. Last Friday Congress passed a law saying that these companies - Equifax, TransUnion, what's the other one, Experian - couldn't charge for credit freezes anymore because they've been charging for those.

Steve: Nice, yes.

Leo: So you can get one for free now in every state of the union. And they extended the length of fraud alerts from 90 days to a year. So, frankly, that's a punishment because that hits them at the bottom line; right? They can't make money off of you if you've got a credit freeze.

Steve: Yes.

Leo: So I'm telling everybody I know, you can now, if you're worried about this, get a credit freeze or a fraud alert at no cost from all three.

Steve: Yeah. So, okay, just to wrap up this story, almost 20,000 U.K. customers had their names, dates of birth, telephone numbers, and driving license numbers exposed; 637,000 customers had their names, dates of birth, and telephone numbers exposed; 50 million U.K. customers had names and dates of birth exposed; 27,000 people in the U.K. had their Equifax account email addresses swiped; and 15,000 U.K. customers had their names, dates of birth, addresses, account usernames, plaintext passwords, account recovery secret questions and answers, obscured credit card numbers, and spending amounts stolen.

Leo: Wow.

Steve: So that just pissed off the U.K. big-time.

Leo: Good.

Steve: And they lowered the hammer. That 500,000 pounds is the largest fine that can be levied against a company. And, yes, it's a slap on the wrist to a $15 billion company like Equifax. But still, it keeps it in the news. And, boy, you do not want to be a CIO of some other company that does something like this. And I have to say, Leo, at this point, look at what we're covering every week. There ought to be a full-time job in every company of size that has something to protect, where that person's sole job is to scan the update notices of every piece of software that they're using, packages like Apache Struts, looking for important updates to it and then pushing the company to roll out fixes. I mean, they ought to just...

Leo: That's something GDPR mandates is a data protection officer.

Steve: Yeah, yeah.

Leo: In fact, GDPR would have killed these guys if this had happened today; right?

Steve: Oh, yes, yes. I found a nice note as I was going through my email bag from Louis Vincent in Ottawa, Ontario, Canada. The subject caught me: "SpinRite fixes Task Manager." And of course that's not actually what happened. But he said: "Hi, Steve. SpinRite owner since 2007. Security Now! listener yada yada. This past week my laptop started grinding to a halt. With no program running, Task Manager would show that the CPU was pinned at 100%." He said: "Did I have a crypto miner on my device, I wondered? The weird thing was that Task Manager was showing that the process taskmgr.exe was itself taking 40-50% of the CPU, with McAfee taking most of the rest." Actually, I have a theory I'll share in a second. "That was odd, at least for the taskmgr.exe."

He says: "I did not find the issue while in Safe Mode. But every time I logged back in normally, taskmgr.exe was back hogging the CPU. I decided to SpinRite the 500MB hard drive. As you might imagine, a couple of hours later SpinRite reports nothing crucial found on the drive. But a reboot later, and everything is working as it was at the beginning of the week. Thanks for a great product."

So this is another story, actually we covered one while you were on your vacation, Leo, about some guy whose security camera system was booted from an SSD, and it was freezing all the time, but not notifying him. So every time he checked on it, it had frozen, and he didn't know when, but all he was doing was rebooting it. Finally he ran SpinRite, and of course no more freezes, which was fortunate because some time afterwards some bad guy was caught on camera stealing stuff from his front yard, and he was able, thanks to the fact that it hadn't crashed, to provide the video to the local sheriffs, who arrested the guy, and he was now behind bars.

Leo: Nice.

Steve: So again, SpinRite didn't show that it fixed anything, but it did fix whatever the problem was. And we've talked about how that can often be the case. My guess is that McAfee was doing some routine background scanning of the system, trying to do its AV stuff, and hitting some spot on the SSD that was giving it some heartburn, or it was reading the data wrong or whatever. And so even though SpinRite didn't raise any flags, because we know that error correction is not perfect, it's sort of like parity, "parity" meaning that there's an extra bit which forces all the bits to have even parity. If one bit is wrong, then the parity is incorrect. If two bits are wrong, then the parity is correct again, even though they're two wrong bits.

Well, it turns out that error correction is good in the same way, but not perfect. And SpinRite is able to not get fooled that way and essentially able to correct problems that the drive itself doesn't even see. And so it often does that, doesn't report anything because it and the drive fixed the problem, yet the problem goes away anyway. So a nice side effect of running SpinRite on a drive where something seems a little flaky, and then the flakes are gone afterwards.

And the last story that I wanted to share is a tip of the hat to Cloudflare. We've got a bunch of friends over there. They're doing great work. We've covered that they did the 1.1.1.1 DNS service. They're offering DNS, both over HTTPS and over UDP TLS, known as

DTLS. So they're very privacy and security conscious. So that allows, if you have a DNS client which is able to do DNS over HTTPS, and we're seeing some web browsers that are beginning to offer that option, the point being that, if you're really concerned about privacy, it's one thing, as I was mentioning before in a different context, it's one thing to encrypt your communications with a remote website. But if somebody is sniffing your traffic and seeing the DNS queries that are going to your DNS server, well, they know where you're going still. They may not be able to see what you're saying to that person, but as we know it's very powerful when, for example, say that you're a bad guy and trying to keep your network of other bad guys secret.

Well, if law enforcement is able just to see, to build a graph of connectivity of who's talking to who, that can be leveraged into some very powerful use. And similarly, if you are an oppressive government that is looking at who some known dissidents are talking to, well, even though they're not able to see what you're saying to each other, they're able to uncover a network. And there are many valid uses for wanting to keep the metadata of your communications private. DNS naturally, because it's not an inherently encrypted protocol, leaks that.

Okay. So what do you do? You encrypt your DNS. There's still a problem, though. And this is a problem that has not yet, until tentatively just now, been solved, even with TLS v1.3 that's the latest version of TLS. And that is the so-called SNI, the Server Name Indication. The idea is this is what's necessary for shared hosting environments. Over the years we've talked about how browsers establish an encrypted connection to a remote server. Normally what happens is the browser will look up the IP address for the service based on its domain name; will then connect to that IP address. And the way originally SSL and now TLS, the evolution of SSL work is that the server sends to the client its signed certificate, signed by a certificate authority that the browser, the HTTPS client, trusts. So since it trusts the CA because it's in the root store, it trusts the CA's signature that the CA has done due diligence in verifying the ownership of the certificate. And so that's how this is established.

The problem is that the certificate is sent to the client immediately after the TCP connection is negotiated as the first thing that the server sends in order for them to establish an encrypted connection. So how do you handle multi-hosting, where there may be many different domains, all accessible at the same IP? Well, for a while there were still browsers that did not understand SNI, that were unable to specify the server that they wanted to connect to. And there were servers that didn't understand it.

Well, we're past that now. All browsers that are in use are able to specify, as an addition to the information they're initially sending, the host name that they want to connect to. So the browser sends its so-called Client Hello packet, which in the old days just said hi, here's a nonce that I've just come up with for our communication. Here's the list of ciphers that I support. Take a look at that and send me back the Server Hello. Now that Client Hello, in an extension field, is able to say, "And this is the server I want to talk to at your end." And that allows then this whatever it is that is answering the connection at the remote IP, it's able to look at the Client Hello, find that extension to the protocol, see the domain that the client is asking for, all among many that live at the same IP, and then select the certificate to use in the Server Hello to send back. So that's how we handle encrypted communications and the disambiguation of which service we're asking to connect to at a single IP. Once upon a time it was just by IP. But in shared hosting environments, that was a problem.

Okay. But because the client and server at that stage have not yet established encryption, we still have information leakage. And that has been discussed, I mean, extensively. And no one has come up with a solution that everybody likes. And the point is that that Client Hello is not encrypted and can't be encrypted, obviously, in an obvious fashion, because the server needs to be able to decrypt it. And the server needs to be

able to decrypt it without any knowledge of the client in advance because otherwise somebody listening on the wire could also decrypt it, would be able to be in the same position of decryption. So that argues that the client has to have some information that works with something only the server has.

And what a small group of developers - Eric Rescorla, whose name I often see in Internet RFCs, he's at RTFM, Inc., which is his consultancy; a developer [Kazuho Oku] at Fastly; Chris Wood at Apple, who's also involved in Apple security stuff; and, not surprisingly, Nick Sullivan of Cloudflare - they got together and authored an RFC proposing a means of solving this problem. And Cloudflare has brought it up, and it's running sort of in a test mode. I mean, there isn't yet a large base of clients to connect to. But they're solving the chicken-and-egg problem by saying, well, we're going to put this online and invite people to connect to us.

And what they've done is clever. They've defined a record for DNS which provides the public key for the encryption of this information by the client that wants to protect the privacy of the name of the host to which it wants to connect in a multihosted environment. So a client that is aware of this extension to TLS 1.3, in the same way, probably at the same time that it does a lookup for the A or the AAAA record, which is the IP address of a site, could ask for the text records that also match that domain name. One of the newly defined text records is a means of encoding a public key.

So the multihosted site has placed a public key for which only it has the private key - notice we don't need any CAs in this model because it's using DNS to do this. And DNS is not yet safe enough to be used in place of CAs for trust. But in this model it provides the security guarantee that we need. It just provides a means of posting a public key which a client can then use to encrypt something that it's going to send to its target. And no man in the middle, no one listening to the traffic is able to crack that because they would not have the matching private key that never leaves the server.

So then the multihosted system sees a new type of extension because, in the same way that there's a Server Name Indication extension to the Client Hello, there's now an encrypted SNI, ESNI extension which these guys have defined, which has been encrypted under the public key which DNS is making available through a text record. They encrypt under that, send the Client Hello packet to the service, which is then able to decrypt that encrypted SNI extension, see who it is the client wants to connect to, select the proper certificate, and send it back.

So just, again, a tip of the hat to Cloudflare and Apple and Eric and Kazuho Oku at Fastly, the guys who put this RFC together, and to Cloudflare for bringing an implementation up and running. This was an actual announcement of the availability of this. Oh, and by the way, this coming Thursday, in two days, is Cloudflare's eighth birthday. They are eight years old in two days and just going a great job on the Internet for us all.

Leo: I agree. Well, and so are you. And this concludes this edition of Security Now!. Thank you, Steve. You can watch us do this show live every Tuesday, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. You can watch at TWiT.tv/live. If you want to download a copy, Steve's got them all at GRC.com. While you're there, check out all of Steve's great free work and, of course, his bread and butter, SpinRite, the world's best hard drive recovery and maintenance utility. GRC.com. He also has transcripts, which is nice. A lot of people like to read along.

Steve: I was joking with Jason over the fact that SpinRite fixes SSDs.

Leo: Yeah?

Steve: And I said, you know, I don't know what I'm going to do because they don't spin. And he said, "Well, call it AllRite."

Leo: Ah.

Steve: I was like, ooh, my god, that's good.

Leo: AllRite. AllRite, AllRite, AllRite. Get Matthew McConaughey to do the ad. AllRite, AllRite. You can get copies of the show, audio and video, from us at TWiT.tv/sn. Or subscribe on your favorite podcatcher, and you'll get it every week, right after the show, soon as it's available. We've got to edit it a little bit, put some stuff at the beginning, put some stuff at the end, but then you'll get it. Steve, have a great week.

Steve: Will do. Glad you're back, my friend. And we'll talk next week, with the one-day warning for the Presidential Alert coming to all of our phones on Wednesday in the early afternoon East Coast, late morning for us on the West Coast.

Leo: We've got to get the word out because people are going to freak out.

Steve: Actually, you probably want to be somewhere public.

Leo: Oh, yeah, because you'll hear it all loud and clear.

Steve: Yeah, that would be cool. Neat.

Leo: Yeah. All right, Steve.

Steve: Okay, buddy. Talk to you next week. Bye.