SECURITY NOW!

Transcript of Episode #623

Inching Forward

Description: This week we discuss and look into DigiCert's acquisition of Symantec's certificate authority business unit, LogMeIn's LastPass Premium price hike, the troubling case of Marcus Hutchins's post-Defcon arrest, another instance of WannaCry-style SMBv1 propagation, this week's horrific IoT example, some hopeful IoT legislation, the consequences of rooting early Amazon Echoes, the drip drip drip of WikiLeaks Vault 7 drips again, Mozilla's very interesting easy-to-use secure large file encrypted store and forward service, the need to know what your VPN service is really up to, a bit of errata and miscellany, and some closing-the-loop feedback from our always-attentive terrific listeners.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-623.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-623-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson's here. There's no big - nothing, nothing urgent. Just relax. We're just inching forward in the security world. He will have his comments, though, on the WannaCry hacker who was arrested on his way home from DEF CON and a whole lot more. It's coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 623, recorded Tuesday, August 8th, 2017: Inching Forward.

It's time for Security Now!, the show where we cover the security and privacy of all of us online, in the real world and the unreal world. And that guy right here, the guy looking over my - peering over my shoulder...

Steve Gibson: "Cyber world," I think, is the official term.

Leo: Cyber. Cyber. Cyber is always hard.

Steve: Yeah. So...

Leo: I didn't introduce you. Steve Gibson.

Steve: Oh. That's me.

Leo: That's Barron Trump, ladies and gentlemen. He is a genius with the cyber.

Steve: Has suddenly grown up.

Leo: All grown up.

Steve: And has lost his hair. And what there is...

Leo: Steve Gibson of GRC.com, genius fellow.

Steve: ...is gray.

Leo: Gray, gray, gray.

Steve: So no real title for this week stood out. So I just thought, okay, you know, how about we'll call this one "Inching Forward" because there are signs of progress amid many indications of need for progress. So for our August 8th Episode 623, "Inching Forward." We're going to discuss a bunch of fun things. The top two were, like, clearly dominant in our listeners' minds because they know that they've both been a focus of this podcast in various ways. The first is DigiCert's acquisition of Symantec's certificate authority business unit. Then of course we have LogMeIn's LastPass Premium price doubling. And I heard you talking about it on Sunday, we need to talk about it, too, the troubling case of Marcus Hutchins's post-DEF CON arrest. Another instance of the WannaCry-style SMBv1 propagation, which was bound to happen.

This week's horrific IoT example gives me an opportunity to talk about the way it should be done, as opposed to the way they did it in one particular case. We have some very good-looking, well-crafted IoT legislation on the horizon. I will be surprised if - and this was introduced by the Senate. I'll be surprised if it just doesn't become quickly signed into law. I don't know why it wouldn't be unless there's some very strong anti-IoT security special interest lobbying. I guess we'll find out. The consequences of rooting early Amazon Echoes. The drip drip drip of WikiLeaks Vault 7 drips again.

We've got a very interesting, easy to use, secure large file encrypted store-and-forward service being offered by Mozilla. And I've taken a look at it, and we'll talk about that. And, I mean, we'll be recommending it. Also the need to know what your free VPN service is really up to. Then a bit of errata, miscellany, and some closing-the-loop feedback from our always attentive terrific listeners. So another great podcast...

Leo: A jolly, jolly good day today.

Steve: ...will two hours from now be in the can.

Leo: Don't rush us. It's going to be a fun ride.

Steve: So our Picture of the Week is actually from last year's DEF CON and Black Hat conferences, but it's just so perfect that I used the reoccurrence of this. Someone sent it to me again. I recognized it, as you did, from having been at a prior conference. It shows the GSM cell providers, both before and then during Black Hat and DEF CON. And it's a little troubling because you pretty much see all of the same ones that were there before, with maybe three times more have suddenly appeared, which is extremely suspicious.

Leo: What do they call those, those phony cell sites? There's a name for them. Oh, Stingrays. Those are Stingrays; right? All the ones that aren't real.

Steve: Yup, exactly, yeah.

Leo: Geez, and there are a lot of them.

Steve: Just wonderful. So, yes, one wants to be very careful when you're doing anything, when you bring any electronics or WiFi or anything to Black Hat.

Leo: Man.

Steve: Okay. So our top of the news was the most tweeted item to me in the past week because everyone knows of my choice some years ago and my continuing happiness - and I should also mention, not only mine, but many people who have sent feedback, some of which I've shared on the show, about DigiCert as my certificate provider. Both Symantec and DigiCert produced press releases over the last week saying that DigiCert and Symantec had some to an agreement about the sale of Symantec's certificate authority business unit to DigiCert.

So before I comment I want to be absolutely clear with everyone, I have no inside knowledge of any sort about this transaction. I spoke to no one at DigiCert or Symantec about this. So any and all conclusions are, you know, they're on a level playing field with everybody else's. I don't know anything. And I have to put that caveat out there because I'm going to assume a few things that are probable.

So all of our listeners know I could not be more bullish on and delighted with the quality of DigiCert's services and their support. And as I mentioned before, our long-time listeners will recall that I was driven away from Symantec and VeriSign and into the arms of DigiCert because I was so unhappy with Symantec's performance. They just, you know, frankly, they were too big. They had a "no way to get their attention" sort of feeling. And I was being upsold, and it just - and they were also very expensive, and it wasn't clear that I was getting my money's worth.

Also our listeners know that DigiCert has been uniquely able to meet some of my own weirder needs, such as minting a pair of specific year-end expirations with specific signature hash strengths in order to satisfy Chrome while Chrome was sunsetting SHA-1 certs, yet I wanted to keep GRC available until that actually happened at midnight on

New Year's a couple years ago. There's no possibility that I could have ever obtained that kind of service from Symantec. Okay, so I'm not going to pretend to be an unbiased and objective observer, although my biases are public. And I believe they're well-earned and deserved. And I'll also note that, if it had been the other way around, if Symantec had purchased DigiCert, much as LogMeIn purchased LastPass...

Leo: It would be time to panic.

Steve: Which we'll be discussing next, yes. I would be heartsick. I mean, I would just - I would be devastated right now since being able to work with and depend upon a high-quality certificate authority is way up there in my own hierarchy of needs, probably only second to Level 3 being GRC's bandwidth provider. And so both of these guys just, I mean, you know, zero trouble service. So anyway, thank goodness the acquisition was in the direction it was.

Leo: Yeah, because DigiCert you use and I use. That's what our TWiT cert comes from. And they're fantastic. We love them.

Steve: They are. And, I mean, they have a boutique feel to them. They don't feel big. I mean, I have email exchanges with them, and I know it's not just me. I hear from other listeners who they know less well, you know, yeah, we're like, wow, I can't believe the support. And, I mean, when I don't interact with them, their facilities that they have established work right.

For example, I use EV certificates, and rather than waiting until it's necessary for me to renew an extended validation cert, which requires a lot more hoop-jumping, their system will in advance start the process of reverifying all of the extra stuff that EV requires. And so my EV-ness is maintained, allowing me to issue my own certificates in the middle of the night. In the dead of night I can get an EV cert in a couple minutes, which is just amazing. And, I mean, but without any reduction in security. They've just figured out we shouldn't wait till the last minute. We should do this ahead of time. And so they just take care of it. I mean, it's just an entirely different experience than I had with Symantec.

Leo: And, yeah, you moved there from VeriSign; right?

Steve: Yes. Yes, I said, okay. And we know there's a lot of switching inertia. I mean, I'm also in the process of leaving Network Solutions, where I have always been. GRC.com is finally getting close to renewal, and it'll be moving to Hover. I'm moving everything over to Hover in the same way that I have already moved everything away from Symantec. So, you know, it takes effort to do it; but it ends up, I think, being worthwhile.

So, okay. So get this, though. There's some money changing hands. DigiCert purchased Symantec's entire certificate authority business for - okay, I won't ask you if you're centered over your ball anymore, Leo, because I know you're in a chair, so hold onto your armrests - $950 million in cash.

Leo: Well, it's not quite a billion.

Steve: Just shy of a billion dollars in cash, plus a 30% stake in DigiCert.

Leo: Oh, wow. Now, that I'm not so crazy about.

Steve: Well, I feel the same way. But, okay. So it must have been a win-win. I mean, okay. So in email which DigiCert's customers received - and this was last week when the deal got closed. John Merrill, who's the CEO, wrote - and I've skipped all the opening boilerplate because the interest to us says, you know, so dot dot dot: "Also, some of you may be wondering about any implications our announced acquisition will have on the ongoing debate between Symantec and the browser community about trust in their certificates."

Leo: Right. See, that's what I wonder, if this is why this happened. It's like...

Steve: Exactly, Leo. And that's why I put that caveat upfront about I don't have - I have no insider knowledge whatsoever. But an observer seeing that - okay. So anyway, I'll finish what John said. John said: "Earlier this year, the browsers proposed a plan to limit trust in Symantec certificates after discovering issues with how they were validating and issuing digital certificates. Importantly," he writes, "we feel confident that this agreement will satisfy the needs of the browser community. DigiCert is communicating this deal and its intentions to the browser community and will continue to work closely with them during the period leading up to our closing the transaction."

And, by the way, that's not until - this doesn't actually close until third fiscal quarter 2018. But I don't know whose fiscal quarter, nor when that is. So it's at least half a year, I would guess, depending upon when the fiscal calendar is, and whose. Anyway, he says, "DigiCert appreciates and shares the browsers' commitment to engendering trust in digital certificates and protecting all users."

And separately, the principal security analyst at 451 Research, Garrett Bekker, wrote: "Symantec's certificate business will immediately increase DigiCert's market share and make the company one of the biggest players in the PKI (Public Key Infrastructure) and SSL markets. This will make DigiCert pretty much one of the leaders in terms of revenues in the digital certificate business."

Okay. So in retrospect, this makes so much sense. As we've talked about, as we've been discussing the various certificate authorities who have screwed up in the past and the reaction of the browsers, which is, I mean, it has to happen. A certificate authority lives and dies by its attention to detail and the earned trust in the integrity of the assertions made by its signed certificates. I mean, that's it. As I've said before, I saw somebody quoting me in Twitter, we're paying them for digital bits. They're minting bits which have value only because of what their signature represents.

And as we know, trust is, as it should be, hard won and easily lost. And Symantec lost the industry's trust as a consequence of now very well documented misbehavior on third-party partner parts that Symantec had a responsibility to manage and didn't. So having acquired their business from VeriSign, which is where they got into the CA business, Symantec was a huge going concern with a large customer base which, unfortunately, as a consequence of these problems which became public, they could no longer effectively leverage. So this transfer of trust-requiring assets from a now untrusted owner who could no longer effectively leverage them for profit to a highly trusted owner who can makes total sense.

So anyway, it'll be interesting to watch how DigiCert handles this. I want them, selfishly, I would love them just to stay exactly the way they are because they're perfect right now. But on the other hand, the certificate minting business is inherently scalable. And as I was saying before, DigiCert's system is already highly automated, and they've just got this nailed. So in theory they should be able to acquire and transfer Symantec's certificate customer base to themselves without needing to massively expand.

So, I mean, I think it's perfect. Symantec had destroyed their trust, and suddenly this asset that they had had crashed in value. And only by moving it, only by transferring it to someone who could effectively leverage its value because they were so highly trusted, could Symantec get the value from it, like obtain its residual value. And DigiCert gets to take advantage of the fact that they can purchase it for a lot less, probably, than they would have been able to before. I mean, it probably wouldn't even have been on the market before. So, you know, yay. I think it makes a lot of sense.

Leo: But there are other acquisitions maybe not so great.

Steve: Well, yes. Speaking of which, all of our listeners were worried. And you and I, Leo, were cautiously, well, optimistic about LogMeIn's purchase of LastPass. We jumped onto the LastPass bandwagon when it was still wet behind the ears. Joe Siegrist, a friend of the podcast, was always forthcoming, shared with me the details of the protocol, exactly how it operated, which is why I was able to explain it to our listeners and say these guys nailed it. They did it exactly right. This is what I'm using moving forward.

And that was even early in the growth of the password manager business. I mean, LastPass was not the first. But I would argue that they had the right set of features, cross-platform. The security model was correct. They did everything they could to secure our data for us in a 100% TNO-obeying fashion. And, I mean, absolute TNO, as we know, comes at a price of convenience. And so they've also cleverly implemented things like pre-issued, one-time, get-out-of-jail-free cards or tokens and things to take a little bit of the edge off of the absoluteness of TNO because they also want to produce a practical solution.

Anyway, and we've followed this now since the LogMeIn purchase. And there's been no indication that we've seen of any failure in the technology. Even when Tavis had his epiphany in the shower, you know, he had barely gotten himself dried off by the time that Joe and company had fixed the bug that he had come up with. Which is all we could ask from anyone is a set of policies that are correct; and then, if mistakes are found, they're immediately fixed. And even in this case of what Tavis most recently found, I mean, it was an obscure problem that it's good to have removed, but nobody got hurt from it.

So, okay. So what happened which has caused a great deal of upset is that LogMeIn - okay. So one way to put it is they've doubled the pricing of their premium subscription. The other way to put it is they increased it by a dollar a month. So it was $1 a month, $12 a year. Or, well, yeah, or free, so there is a free version. But the $12 per year is now $24 per year. There have been a class of users who were willingly paying the dollar per month to support Joe back in the beginning. And probably inertia and the fact that LastPass's value at $1 per month for the premium subscription with the additional features it offers, I think it's a bargain.

So I think it remains to be seen how those people will feel about this jump. I've seen some people saying, oh, my god, you know, I've got to go find a different password manager. Or saying, hey, I was supporting Joe at a dollar a month. This rubs me the wrong way. The free version, it's important to note, remains fully functional and useful. And so I guess I would take a look at it objectively. Rather than being upset at the change, I would say, okay, what are the alternatives? How do they compare? What are their track records? For me, what I care most is about the security guarantees that LastPass offers and the quality of their service, that is, going forward. As with the CA business, trust is hard-earned and easily lost. And so far Joe and company, even after the LogMeIn acquisition, have never let us down.

So I'm not switching. I still think it's the right solution. But I certainly just wanted - because many people were wondering what I thought. You know, I mean, I don't know what's going on behind the scenes. It would be interesting to know, and we never will, what the effect is for LogMeIn. This is the nature of a parent acquiring something like this is they want to monetize it. And they now have built, thanks to Joe and their acquisition, a leading password manager that I think, independent of its cost or its pricing, is the one I still believe is the one to use.

So I guess I would separate the emotion from the service that's being delivered. And for what it's worth, I'm staying where I am. We know them. We know their focus on the technology. The second that changes, then that's a different story. And if you're a person who was paying a dollar a month, or $12 a year, to support Joe, and $24 seems like more than you want to do, then look at the free version. Look at backing away from premium and continuing to use it. So anyway...

Leo: I don't think 24 bucks a year is very expensive. And it's, I think, still less than 1Password, which is, I would say, the number two competition.

Steve: Correct, correct.

Leo: And they moved from a one-time-only fee to a monthly fee for the same reason, I think. It just makes sense. Why wouldn't you want to pay two bucks a month for a good password manager? That doesn't seem too much to me.

Steve: Yeah, well, I mean, I have a problem with...

Leo: There's free ones. They're just not as convenient. So if you don't - if you want to go free, get KeePass or something like that.

Steve: Yes.

Leo: And give up some convenience. But you can save the money.

Steve: Well, and I have a problem paying a subscription for something that doesn't have an ongoing cost to the provider. So, for example, BoxCryptor, we'll be talking about them later, they used to have a legacy version which they've discontinued. The legacy version you could purchase, and then you owned it, and you could use it on all the cloud providers that you wanted to. To me, that makes sense. Now they only have a pay-as-you-go subscription business, and they lose me. Sorry, I'm not doing that. There are alternatives.

But, for example, in the case of LastPass, we're using their servers. They're maintaining the apps that we use on our various platforms. Joe is there when Tavis comes out of the shower on the weekend. It was a Saturday afternoon, I think; you know? And bang, within an hour this thing was fixed. So that's what we're paying for. So there it makes sense that it would be an ongoing support for what it is that we're getting in return. In this case, I think it makes sense. It's not a standalone utility. It is networked. It's cloud-based. And we need it to be. To be useful, it's got to be cross-platform. And we know they're continuing to put a lot of work into it because we keep seeing improvements.

Okay. Marcus Hutchins, Leo.

Leo: I'm really interested in what you think. I mean, I think we don't have enough information to know. Is he a criminal? If he's a criminal, he's a criminal.

Steve: No.

Leo: Or is this overreach by federal prosecutors using hacking laws that are antiquated and not well defined? Or is he completely innocent, and they're just being jerky jerks?

Steve: So let's catch our listeners up, and then we'll talk about this. Of course Marcus came to everyone's attention in May, a couple months ago. He was the guy who stopped the WannaCry worm by registering that obscure DNS domain that it was checking for. And he discovered, quite to his surprise, that it suddenly stopped propagating. So this crazy SMBv1 weaponized EternalBlue exploit that was WannaCry got stopped in its tracks because this white hat hacker reverse-engineered enough of the worm to spot its attempt to retrieve a web asset from that bizarrely named and, at the time, unregistered domain.

And, okay. So last Wednesday, six days ago, as Wired wrote in their coverage, authorities detained - and they said 22-year-old Marcus Hutchins, although Ars Technica's coverage said he was 23, and they included a screenshot of his booking, and I had it - oh, at the City of Henderson near Las Vegas - that did show his age as 23. So I think they're probably right. Anyway, this was immediately following the DEF CON hacker conference last week in Las Vegas, as he was attempting to fly home to the U.K., where as we know he works as a researcher for the security firm Kryptos Logic, spelled K-R-Y-P-T-O-S.

So upon his arrest, the Department of Justice unsealed an indictment against Marcus, charging that he created the Kronos banking trojan, which is a widespread piece of malware used to steal banking credentials for fraudulent purposes. He's accused of intentionally creating that banking malware for criminal use, as well as being part of a conspiracy to sell it for $3,000 between 2014 and 2015 on cybercrime market sites such as the now-defunct AlphaBay dark web market. Now, it's worth noting that on Friday, two days later, when Marcus was arraigned, he denied any wrongdoing and pleaded not guilty to the charges against him. He has a court date set for today, August 8th, in

Milwaukee. And until then, until probably just being transferred, he was being held in Las Vegas jail.

Now, it's also worth noting that his alias is unfortunately "MalwareTech," and he uses the Twitter handle "@MalwareTechBlog," which is probably not helpful, but that's who he's known as.

Leo: And not a crime.

Steve: And not a crime, exactly. So paraphrasing a bit from Wired's coverage, the short, eight-page indictment against Hutchins has already raised questions and skepticism in both legal and cybersecurity circles. Orin Kerr, a law professor at George Washington University who has written extensively about cybersecurity and hacking cases, says that based on the indictment alone, the charges look like, quote, "a stretch." Although the indictment claims Hutchins wrote the Kronos malware, nothing in the document illustrates that Hutchins possessed actual intent for the malware he allegedly created to be used in the criminal conspiracy he's accused of. Kerr said, quote: "It's not a crime to create malware. It's not a crime to sell malware. It's a crime..."

Leo: Really? Wait a minute. It's not a crime to sell malware?

Steve: That's what he said. That's what this guy says, yeah, which surprised me, too.

Leo: It's a crime to use it, I guess.

Steve: Well, he says: "It's a crime to sell malware with the intent to further someone else's crime." Meaning that, like, if you sold it to your grandmother because she wanted to hang it on the wall, then that's not a crime. But clearly most malware sales is with the intent to further someone else's crime, so...

Leo: It's hard to imagine, I mean, you could create malware for research purposes.

Steve: Right.

Leo: But selling it, you wouldn't sell it to another researcher.

Steve: Yeah, no.

Leo: Now, that's, you know...

Steve: But so I guess the point is...

Leo: He never sold it, though, I guess, and that's part of his defense is he made it, and somebody else he either knew or didn't know, it's not clear, sold it.

Steve: Yeah. So anyway, Kerr says the story alone doesn't really fit. There's got to be more to it, or it's going to run into legal problems. Now, of course, there are a lot of people who are behind Marcus and are supporting him. Some associates of his defended him Thursday on Twitter, that's last Thursday, even arguing that he has worked directly with U.S. law enforcement. Kevin Beaumont, whom we've spoken of before, who is a U.K.-based security architect, wrote: "I know Marcus. He has a business which fights against exactly this, bot malware. It's all he does." Kevin said: "He feeds that info to U.S. law enforcement." And he wrote: "The DoJ has this seriously F'ed up," and he didn't say "F'ed" in his tweet.

Jake Williams, another well-known researcher with the security firm Rendition Infosec, said he'd worked with Hutchins multiple times since 2013, met him in person at last year's DEF CON, and shared malware samples. At one point in 2014, Williams says, Hutchins refused his offer of payment for help on an educational project. And as we know and reported, even when Hutchins was awarded a $10,000 bug bounty from the security firm HackerOne for his work on stopping WannaCry, he donated it to charity. Williams said: "I have pretty good black hat radar. It never went off when talking to Marcus or exchanging stuff with him."

So Wired finishes the coverage writing: "For the moment, neither the FBI nor the Department of Justice is commenting further on Hutchins's case beyond the DoJ's statement and the facts of the indictment." A spokesperson for the Electronic Frontier Foundation, our EFF, which often offers legal representation to embattled hackers, wrote in a statement to Wired that it's "deeply concerned" about Hutchins's arrest and are reaching out to him. In their tweet, the EFF tweeted on August 3rd: "EFF is deeply concerned about security researcher Marcus Hutchins's arrest. We are looking into the matter and reaching out to Hutchins."

And in the show notes there is now a legal defense fund that has been set up at secure.lawpay.com. I've got the link in the show notes, and the show notes are already posted online, so anyone can get them at GRC.com/sn. And I scrutinized this page carefully. It looks legitimate and looks like the real deal. So I wanted to let people know, if anyone is interested in contributing, it looks like this is the place. You might want to wait until the EFF puts something up on their site. I would trust, clearly, a link from them. I don't remember where I got this link. It was part of the research that I was doing putting this together. But it does look legit.

And it'll be interesting to see how this turns out. And your coverage on TWiT on Sunday, Leo, I think, was spot-on. Your panel agreed with what we were saying in fact just last week about the problem with this kind of reaction is the chilling effect it has on the benefit that white hat hackers produce for society at large in finding these. I mean, clearly he really helped minimize the problem with WannaCry because he felt it was safe to do this. And in fact you guys were talking about the fact that there are other security conferences, and how would out-of-country, non-U.S. researchers feel about coming to Black Hat and DEF CON if they thought there was some fear that they might get nabbed by border security, essentially, when they're trying to leave.

Leo: Yeah. And of course a lot of the opinions people have on this are preconditioned by how poorly the DoJ has acted in cases like Aaron Schwartz's case, where they overreached, and Marcus's reputation in the hacker community. So while we don't know the facts, there's I think legitimate reason to be concerned and to want to know more.

Steve: Yeah. And we've also, you and I have talked on this podcast about how it is also the case that some hackers sometimes do cross the line. That is, you know, four years ago he may have been a different person from who he is today in 2013.

Leo: Right. We just found out, yeah.

Steve: And so, you know, he was younger, less experienced. And it's also the case that a lot of sort of more junior hackers start off a little more on the dark side and then realize, hey, I can do good with this rather than just bad. So anyway, the good news is he will probably have access to the best defense that he deserves. Let's hope he deserves a really good one, and then it will happen.

Leo: Good, good.

Steve: So in a follow-on, and under the category of, well, this was entirely foreseeable, researchers at Flashpoint have discovered that the so-called "TrickBot" banking malware trojan has just been evolved to add WannaCry-style SMBv1 LAN scanning and propagation. It doesn't yet have the capability of scanning out into the public Internet, which was WannaCry's big claim to fame. But it does probe LAN-based servers via there's an API in Windows, NetServerEnum, that essentially allows software to get a list of servers that are known to be on the network. And so it gets that list from Windows itself and then says, oh, thank you, and then does a WannaCry-style SMBv1 attack against them in order to insert itself and propagate through the Intranet within an environment that has servers. It can also use LDAP, the Lightweight Directory Access Protocol, in order to find candidate servers to attack.

So if it gets inside an enterprise network which has not yet administratively and globally disabled SMBv1 - yeah, which I'm hoping all of the IT managers who listen to this podcast did back in May when Marcus Hutchins was shutting WannaCry down, and we found out that this was an SMBv1 vulnerability - and nobody needs SMBv1 - and noticed that it won't be until Windows 10 later this year that Microsoft disables SMBv1 proactively.

So it's up to IT people in enterprises to do that now. You don't want this thing getting loose inside your network. And it's also foreseeable that this won't be the last re-use of this SMBv1. If history teaches us anything, we know that it's going to take a long time for SMBv1 to finally disappear, even though it's been deprecated for decades. I mean, it's more than 20 years old. It was the original file and printer sharing protocol in Windows 3. So, yeah.

Oh, boy. And in this week's IoT nightmare, we learn from the guys at BitDefender that they found at least 175,000 publicly available, publicly exposed, Chinese manufactured, Internet-connected security cameras located and indexed by Shodan which are completely hackable. So the guys at Bitdefender, doing some poking around with Shodan

- we've talked about it before. Shodan is a queryable search engine for all the other ports on the Internet, rather than 80 and 443. Eighty and 443 are of course HTTP and HTTPS, which Google indexes for us. Shodan does the rest. It doesn't deeply index the content the way Google does, but it scans the entire Internet and just says, oh, I found something over here on port 32726 - wait, 327, yeah, it's valid - and then allows you to make a query, if you have something that you want to find that you know lives on that port, and off you go.

And it also, if you have a server that responds to a connection, it indexes the response. So, for example, if it's a web server in a camera, and that web server in the camera says hi, I'm from Shenzhen Neo Electronics, the NIP-22, which is actually one of the models which is vulnerable, then Shodan indexes that. So if you determine that the NIP-22 has a vulnerability, Shodan says, oh, here's 175,000 of them that are currently on the Internet. Have fun.

So, okay. So these vulnerable cameras are manufactured, as I said, by Shenzhen Neo Electronics. They produce security and surveillance solutions, including IP cameras, sensors, and alarms. Bitdefender did some research, discovering buffer overflow vulnerabilities in two models of the cameras. They have one called the iDoorbell. Apparently it's being sold on Amazon, or there is at least an iDoorbell on Amazon which has one one-star review.

Leo: Yikes.

Steve: I couldn't determine if it was the same one. I couldn't find - I looked all over for the manufacturer. I don't know. But apparently it got like really hot when this one guy plugged it in. And it's like, okay, this doesn't seem right.

Leo: Do not buy that, yeah.

Steve: No.

Leo: It's not the Ring, that's for sure.

Steve: No. This is the anti-Ring.

Leo: Yeah, the opposite end.

Steve: Oh, boy. And the other one is the NIP-22 model. But because many of Shenzhen Neo's devices share and reuse the same firmware, these researchers believe that other models also are certainly similarly vulnerable. The security cameras - now, get this, Leo, here's the ultimate nightmare - use UPnP, Universal Plug and Play, to automatically open ports through their users' firewall and router to allow unsolicited incoming access from the public Internet.

Leo: Geez Louise.

Steve: Yeah. Querying the Shodan search engine for vulnerable devices, the researchers discovered, depending upon when they queried, between 100,000 and 140,000 vulnerable devices of various sorts, leading them to sort of conclude, okay, about 175,000 is what we're guessing, if you remove various overlaps.

Okay. Now - oh, and there is both an HTTP and an RSTP. That's the real - RTSP, sorry, Real-Time Streaming Protocol. Both services and servers are exposed. Okay. So this is the least caring, least careful, and sloppiest way of doing this, by simply creating Internet-exposed, publicly available TCP servers. So as a lesson, the proper way to do this for such devices, which are almost certainly going to be located behind NAT routers, is for those devices to occasionally send a single UDP packet outbound to a publicly accessible remote server.

The flow, the travel of that outbound UDP packet automatically creates, I was going to say "instantiates," same word, fancy word, creates return packet mapping through any and all NAT routers it encounters and passes through along the way, which denies incoming traffic from any but that single remote IP. In other words, traffic coming back is not unsolicited, but any traffic coming from any other source IP can't get through. It'll just be dropped, as it should be.

Then the remote server can use the implicit reverse mapping through the NAT routers, which is created by the camera very occasionally, just sending a little ping, essentially a UDP ping packet out, that allows the remote server to send instructions back on a per-camera basis when it wishes to enable, for example, audio and video streaming. So that's the way you do this. It's not difficult. It just requires someone who cares. Instead, they just plug the camera in, and it opens up TCP, HTTP, and RTSP services that anybody can access.

Okay. So on top of this, Bitdefender found that both of these camera models are vulnerable to two different remote attacks, one that affects the web server service running on the cameras and the other that affects the RTSP, Real-Time Streaming Protocol server. They showed that it was quite easy to exploit the flaws in the security cameras. This doesn't even require high-end hacking. Get a load of this.

First of all, they've got default credentials, believe it or not, user/user and guest/guest. So, yes, the username is "user," and that password is "user"; and the guest account the username is "guest," and its password is "guest." So then they also found several buffer overflow vulnerabilities that could be exploited to take control of the cameras remotely. And again, those are trivial.

An example of one of the vulnerabilities is that, in the publicly exposed web service, it's triggered by an error, that is, the overflow is triggered by an error in the way the application processes the username and password information. When the user attempts to authenticate, the credentials are passed in a GET request with name=value parameter pairs, so it's the <ip>/?, so then usr= and then the username, ampersand, pwd= and then the password.

But when the web authentication function parses these values - that's the libs_parsedata function - it copies the content of the two arguments onto the stack without checking their actual size. You know, you just can't make this up. Thus, of course, allowing for an out-of-bound data write onto the device's stack, which it's then easy to arrange to have that function execute when it returns, and the remotely provided code is then running in the camera to get up to any mischief it chooses. Although, boy, with default usernames and passwords of user/user or guest/guest and a publicly exposed TCP server, it's not even clear that you need to bother with a buffer overrun.

So, boy. I mean, it just can't get any worse. This is pure irresponsible sloppiness. But at the moment we have no policing of this. Anybody can produce anything and sell it and say, oh, look, Internet cameras, they work. Yeah. But if you have this inside your network, not only can anyone see what's going on through your camera, but they can generate a query that ends up allowing their code to get into your camera, probably giving them access to the rest of your internal network.

So this is a perfect example of a way that an IoT device compromises your internal network because, if it's a WiFi device, it's on your WiFi, so it's got access to your Intranet. And if remote hacker code is able to get in there and establish a beachhead, then they're in your network. And there are 175,000 networks currently exposed to exactly this threat as a consequence. Everybody who bought one of these things and just plugged it in and said, oh, look, I can see what the baby's doing when we're not in the baby's room.

Leo: Me, too.

Steve: Wow. Yeah, exactly.

Leo: We all can.

Steve: And so can Shodan and everybody else. Yikes. Okay. But on the heels of that, there is some very good IoT news. As I mentioned at the top of the show, a new U.S. Senate bill has been proposed which will set security standards for Internet-connected smart devices and will use the U.S. government's buying power as its enforcement side. A pair of well-known U.S. senators, one from each political party because this is not a partisan issue, Virginia's Democratic Senator Mark Warner and Colorado's Republican Senator Cory Gardner have introduced last week a new bill titled "The Internet of Things Cybersecurity Improvement Act of 2017." And they did some homework. I think they had help with - I want to say Harvard. I don't think I wrote it down. But they did get some knowledgeable input, it's very clear.

I read the legislation as a 20-page document. For example, it forbids any interconnected device purchased by the U.S. government from having hardcoded, unchangeable usernames and passwords in those devices. The legislation also requires vendors to ensure that their devices are patchable and, at the time they are purchased, are free from already known vulnerabilities. Firmware updates must have an effective authentication mechanism such as a secure digital signature which is verified to prevent unauthorized updates. The device must use only - this is what the legislation actually says. It's wonderful. The device must use only non-deprecated industry-standard protocols and technologies for communications encryption and interconnection with other devices or peripherals.

The device must be updatable. Quoting from the legislation, it writes: "Requires such Internet-connected device software or firmware component to be updated or replaced, consistent with other provisions in the contract governing the term of support, in a manner that allows for any future security vulnerability or defect in any part of the software or firmware to be patched in order to fix or remove a vulnerability or defect in the software or firmware component in a properly authenticated and secure manner."

And it must be repaired in a timely fashion. The legislation says: "Requires the contractor to provide a repair or replacement in a timely manner in respect to any new security vulnerability discovered through any of the NIST and other relevant security databases or from the coordinated disclosure program."

So, boy, you know, this is what we need. It's not law yet. But as I said, it's difficult to see how this hits an iceberg. It sounds like something that the House would not have any problem adopting. It's a nonpartisan deal, so it really feels like there's a chance that the government is, first of all, you know, legislators are waking up to the threat of IOT. And by placing these requirements on IoT devices, the responsible IoT vendors can easily meet these requirements. I mean, this is not onerous. This is just the way it should be done. It's only because there's been, like the case of the Shenzhen Neo cameras, absolute blatant irresponsibility, the abuse of technology, that this looks like in retrospect a huge change. But it's the huge change we need.

So hats off to Warner and Gardner for putting this together. And, I mean, it's everything in there is what we would like to see in an initial first pass at working to improve things. And you've got to know that the U.S. government is a huge purchaser, and a lot of vendors would like to be able to sell to them. Again, this will require vendors do what they should have always been doing. And as a consequence, we'll see security improve. And, boy, if there could be a certificate program or something so that we consumers could know that this meets the U.S. government purchasing guidelines, and that as a consequence it's compliant with this legislation once it gets pushed into law, that would be a good thing, too. So anyway, as I said, "Inching Forward," the title of this podcast, because this represents some progress.

Leo: [Crosstalk], one step at a time.

Steve: Okay. So rooting an Amazon Echo is now, or at least was for a while, a thing. Mark Barnes wrote last Tuesday, or posted while we were doing last week's podcast, a topic. He used the "A" word. I'll say, "Echo, are you listening?" So this is interesting, but not overly alarming, because it requires physical access - so the so-called "Evil Maid" attack, meaning somebody that has access to your home could get up to some mischief with your Amazon device - and only applies to the earlier 2015 and 2016 model Amazon Echoes, although there are a lot of those that are out there because they were popular back then.

So it turns out - and Leo, on the next page of the show notes I have a picture of the bottom of the earlier Echoes. Actually, it's the Echoes now. They just changed the design a little bit. Under the rubber base of those earlier Echoes, and actually of even the current ones, is an 18-connection, very powerful debugging and access pad which provides, among other things, a serial terminal interface where, if you hook a serial terminal to it and then plug the Echo in, you see all this Linux booting scrolling stuff, and with the names of everything. So, I mean, its complete internal information disclosure just exported out of a ground and the UART transmit line. So there's that, and support for a remote SD card booting interconnect.

Together those two things allow for operational discovery. You're able also to talk to the Echo through the serial input, which allows the device to be monitored, probed, and then ultimately fully commandeered, without opening the Echo any further. You take the rubber base off. You press something up against the bottom, and you're off. So Mark Barnes, who is with MWR Info Security, said that his team developed scripts that leveraged tools embedded on the Amazon Echo to continuously stream raw microphone audio over TCP/IP to a remote server without affecting the usual functionality of the device itself, thus as a proof of concept turning it into an always-on home surveillance device.

Thus, in the fully developed attack scenario, an Evil Maid would disconnect the Echo's power, remove the rubber footing, press a square connector pad against the exposed 18-connection debugging surface - a which would acquire power and access to the internals - then power up the Echo and wait while it boots. The external SD memory would take over, modify and rewrite the Echo's internal firmware on the fly, turning the device into anything the attacker wished, like a 24/7 audio exporting streaming device. Then the boot override would be removed, the rubber base reaffixed, and the Echo returned to its original location, essentially having been modified.

So users owning this year's 2017 models of the Echo are not affected by this latest hack because the new models introduced a mitigation that joins two of the crucial debugging pads in a way that prevents the device from that external booting. So there's still a lot there but you can't do a boot override in order to bypass the internal firmware.

So our takeaway is that this is an interesting attack. But we've seen these before. We'll all remember the Samsung television attack where, if you stuck a USB key into the Samsung television, you could also compromise it. But not remotely. Not over the network. Not from the public Internet. You had to be physically present. So it, too, was an Evil Maid style attack. Arguably...

Leo: Evil Maid with some skills, in this case. She'd have to build that thing.

Steve: Well, yes, but...

Leo: I mean, they didn't offer that for sale; did they?

Steve: No. And, see, but that's the next step. And so, I mean, for example, this is the kind of thing we could see the CIA doing, if they wanted to have access to somebody's audio and got a warrant in order to do so - get into the house, modify the Echo, and then leave. And now it's a streaming device. But again, not anything to get all worried about. It requires physical access. Cannot be done on the newer devices. And, who knows? This is very fresh. It would not be unforeseeable that Amazon would push a firmware update a few days from now that would even disable the ability to override it by SSD, if that's possible to do through a firmware update. I don't know one way or the other. I mean, that little 18-pad connector, that is a very, I mean, that's an internal goldmine of stuff.

Now, again, you could also say, okay, all they did was, that is, all Amazon did by exporting those 18 connections is make it additionally easier. I mean, for example, the Google Home. It may not export all those pins. But if you open it up, all that same stuff is there. And so again, physical access, as we know, always lets you do pretty much anything you want. So I don't consider this a weakness, which is why there's no reason to run around and get concerned. It's just, you know, another example of us having a powerful Internet-connected computer in our homes that, if somebody has access to it, can be subverted. But then all of our computers are that way.

Speaking of the CIA, the drip drip drip of WikiLeaks Vault 7 CIA leaks continue. There was some coverage of the most recent, I guess we're at 19 now, or maybe it's 20, weeks of disclosures. This one's called Dumbo. They all get names. And again, this is another one that's not a huge concern. Dumbo is, if we're to believe that these are valid CIA documents disclosing internal technology, it requires system-level privileges on a Windows machine which is responsible for monitoring streaming audio and video devices in some environment, and physical access. A USB drive must be plugged into the Windows system throughout the operation to maintain control over surveillance devices.

So what does it do? It can mute all microphones, disable network adapters, suspend any processes using a camera recording device, and then selectively corrupt or delete recordings. So as I read about that, okay, first of all, this seems like it's not doing anything fancy like we see in the movies, like recording a minute of video with nobody doing anything, and then looping that video inside so that everyone continues to think nothing is happening while they go tiptoeing around, modifying people's Amazon Echoes. That doesn't happen. It just basically creates a blackout.

So it has the feeling to me of something that was originally purpose-specific. It was created to address some specific need somewhere. Agents needed to shut down an audio/video surveillance facility where a blackout was preferable to being seen and recorded. So the tool was - and I'm just hypothesizing, of course. The tool was commissioned and created for that purpose, but then added to the arsenal of possible tools of some future use. So again, this is not anything like EternalBlue. But it's sort of an interesting bit of arsenal that it looks like, if we're to believe where all this came from, that our law enforcement and intelligence services have access to.

Okay. I'm going to explain about Mozilla's very interesting Send service, and then we'll take our last break. Okay. So the URL is send.firefox.com. And Leo, I'll be interested to know what browsers you have that it works on. Naturally, it works on my Firefox.

Leo: This is Chrome on Linux, and it seems to work.

Steve: Good. What about Safari? Because it did not like mobile Safari. I tweeted that it was browser agnostic earlier today, and somebody sent me a page from it looked like their iPhone where it said, oops, sorry, your browser does not support the required HTTP levels of operation. So it doesn't look like it runs everywhere.

Leo: What does it do?

Steve: It just puts up a page saying, sorry, this browser won't work.

Leo: No, but if it works, what does it do?

Steve: If it works, you should get a page inviting you to drop a file on it.

Leo: Yeah. And then what?

Steve: Well, and that's what I'm going to talk about now.

Leo: It seems to work on Safari on the desktop. I mean, I'm getting the page. I haven't tried dropping a file on it.

Steve: Yeah, I'm not sure where the - my guess is that the moment it came up it said, oh, sorry, we can't work on your device.

Leo: Yeah. Private encrypted filesharing, yeah. I use, you know, I use a command-line service that does the same thing called transfer.sh. But it's not run by anybody as well known as Firefox.

Steve: Okay. So here's what this does. It's very cool. I vetted the technology. It's all open source, all on GitHub. It's another one of the Firefox Test Pilot projects. They do web experiments to sort of explore things. It uses AWS as its backend file storage. So here's how it works. Anybody who - and the way I called it, this is not for content distribution. This is - I call it a "store and forward" because the file stored, it expires after 24 hours or one download. So it's not meant for somebody putting something up for mass broadcast. And I think that's a nice tradeoff because it's a free service.

Okay. So a file, a big file, up to a gig, you're able to drop on the page. It uses JavaScript running in the browser to generate a symmetric random encryption key and encrypts the file. It then gets a unique file identifier token from the server and uploads the encrypted file to the server. So all the server gets is, as we know, when crypto's done properly, is a blob up to a gig, a gigabyte blob of pseudorandom noise that it has no ability to decrypt. The browser then gives you a URL which is send.firefox.com/download/ and then a one, two, three, four, five, six, seven, eight, nine, a 10-digit, it looks like it's hex, file ID. Then a /# and the key, that is, the decryption key for that.

Now, so that's what you then arrange to send to someone. And it's your responsibility to send that securely. Maybe you're just sending it to yourself. You could just write that down because it's not impossible to write it down. The key's not that long, nor is the file ID. So to transfer a big file between machines, or you want to send a blob to a friend of yours, so you drop the file on your browser. It transfers it up. It encrypts it locally, sends it up to the server where it will stay for 24 hours. You then email your friend this key, maybe change a couple digits and then call him and say, okay, here's how you have to fix that, or leave off some or something. So you could figure out ways to make it safer to send through email.

The point is this is not meant to be absolute crazy bulletproof security because of the need to send this URL. But what's cool is, and what's clever, is that - and this is something we've never talked about before. The pound sign truncates the URL in the browser, meaning that nothing after a pound sign is ever sent by a browser that is querying a URL. The IETF calls that - formally it's known as the "fragment identifier." And the IETF spec for the format of a URI says: "When a URI reference is used to perform a retrieval action on the identified resource, the optional fragment identifier, separated from the URI by a crosshatch character" - the pound sign - "consists of additional reference information to be interpreted by the user agent after the retrieval action has been successfully completed. As such, it is not part of a URI, but is often used in conjunction with a URI."

So the point is your buddy or yourself, within a day, on a different machine, essentially goes to that URL with any compatible browser. And it's meant be cross-browser, so it's not Firefox only. The browser will not send the decryption key to send.firefox.com. It only sends the left-hand side, including the file identifier. So that performs the retrieval. The file is downloaded to your browser; and then, exactly reversing the process of it encrypting the file, it uses that tag after the pound sign to decrypt the file, and the recipient is then able to access it. And having done that, the file is then deleted from the remote server.

I played with it a little bit. The crypto looks solid. And again, I'm not wanting to suggest that this is world-class, absolutely utterly bulletproof. But it's all open source. It is inspectable. And it solves an interesting problem. It makes it trivial for, especially, somebody not technically astute. Leo, you were saying you have a command-line system that you use. Here's something you could tell somebody who has a hard time figuring out that Google is not the Internet to say, "Okay, here. Drop that file on this page. You just go send.firefox.com." That's all you would have to tell them. "Drop the file on the page, and then send me the URL you get." And then you'll be able to retrieve the file.

And the key is, it is encrypted in transit, while stored, and can be huge, up to a gigabyte. And it's free. So it gets this podcast's full endorsement. It does everything right, with the understanding that you're responsible for handling that URL responsibly because the decryption key is in the URL. And you can of course split it apart and send it different ways, or dictate it over the phone, or text message it, or do whatever. So a very nice piece of work.

Leo: Yeah, that's cool. I hope they keep it going.

Steve: Yes, as do I. I think, if it is successful, they probably will. Oh, and it's worth noting it is open source. And so anybody who wished to - it is dependent upon AWS. But anybody who wished to could certainly spin up their own AWS instance. And as I have been saying, AWS is surprisingly inexpensive.

Leo: I wonder if they're using the DutchCoders code because that's what transfer.sh is. And it allows you to run it locally as your own server or with S3. And I'm wondering if it's related to that code. Interesting.

Steve: Could be. Although this is not difficult. I mean, I could write this.

Leo: No, it's not a hard thing to do, yeah, exactly.

Steve: Yeah, yeah.

Leo: But that code's open source on GitHub and widely available, widely used. So, yeah, it makes sense to do that.

Steve: Cool.

Leo: Yeah. And, yeah, if you upload the wrong one, just upload another one, upload the right one, and send a better URL. Somebody's saying is there a way to delete that file.

Steve: Oh, yes.

Leo: Is there?

Steve: And in fact I should say - yes. First of all, it self-deletes after being downloaded.

Leo: Right.

Steve: And after you have created one, if you go to send.firefox.com, there's a beautiful managed list of all the files you've created.

Leo: Oh, that's nice.

Steve: And the remaining hours and minutes before their self-expiration. There's a button you can click to manually delete one, if you choose to. I saw online...

Leo: That's much better than transfer.sh. So that's very different. Okay.

Steve: Online, someone said that the key was based on the hash of the unencrypted file. And I thought, that's interesting. And so I verified that by doing a transfer twice of the same file and seeing that the encryption key was different both times. So while it may involve a hash of the unencrypted file, it's not entirely dependent upon it. And in fact involving the hash would help to overcome any limitations or concerns about the integrity or the quality of the pseudorandom number generator that the browser has access to, although that also has been getting a lot better recently. So, yeah, so you end up with a beautiful little file manager also where all the things you've created that have not yet expired are shown by name, size, and remaining time before self-expiration, and the ability to remove it if you so choose. Anyway, very nice piece of work.

Leo: One gigabyte size limit.

Steve: Actually, I think that's a soft limit. The way they put it was "for reliability." So I don't know if it actually has a problem, or if you try to send something bigger it just kind of says, oh, boy, ow.

Leo: [Wordless utterance]

Steve: Inching forward.

Leo: Inching forward.

Steve: So another example of why we need to allow research into the operation of things that their owners would rather we didn't look at. Hotspot Shield VPN, which is a free app and service, has been accused of spying on its users and selling the data. It was developed by a company called AnchorFree. It's a free VPN service available on Windows, Windows Phone, Mac, iOS, Android, Chrome. And unfortunately in this case you get more than you bargain for. They boast that it's in use by more than 500 million users around the world. That looks more like a download count, though, so you can't - it's not clear how many people are continuing to use it. And with any luck, after the news that I'm about to share gets additional coverage, even fewer people will be using it.

The Center for Democracy and Technology, the CDT, is a U.S.-based nonprofit advocacy group who, working with Carnegie Mellon University, carefully examined the Hotspot Shield system and its clients. Yesterday, that is, Monday, the 7th of August, they filed a 14-page complaint with the U.S. Federal Trade Commission alleging that it is willfully violating its own privacy policy of providing "complete anonymity." The Hotspot Shield VPN app promises to "secure all online activities, hide the users' IP addresses and their identities, protect them from tracking, and keep no connection logs, while protecting its users' Internet traffic using an encrypted channel." Which as we know a VPN does when it's done right.

However, according to research conducted by the CDT along with Carnegie Mellon, the Hotspot Shield app doesn't live up to these promises and instead logs all connections, monitors the users' browsing habits, redirects their traffic, and sells their customers' data to advertisers. Hotspot Shield was even found to be injecting JavaScript code using iframes into web pages for advertising and tracking purposes. Reverse engineering of the app's code revealed that the VPN uses more than five different third-party tracking libraries. So it is actively tracking. It's doing far more than providing an encrypted VPN tunnel.

The researchers also found that the VPN app discloses sensitive data including the names of wireless networks via the SSID, along with unique identifiers such as its users' unique Internet MAC addresses and their mobile device IMEI numbers, none of which are formally and properly part of IP traffic, which is what a VPN conveys. So things like the Ethernet MAC and the IMEI are local network numbers, not IP-based numbers. So there's no excuse for those things being captured and transmitted to the other end.

And as if that all wasn't enough, it sometimes redirects ecommerce traffic to partnering domains. If users visit commercial websites, the app redirects that traffic to their partner sites, including ad companies, to generate revenue. For example, the researchers found that when a user connects through the VPN to, in the cases they looked at, www.target.com and www.macys.com, the application intercepts and redirects HTTP requests to partner websites that include online advertising companies.

So, as we know, this is all tempered by the fact that not even a VPN can peer into HTTPS traffic. So some of what they're doing is limited to the traffic that they can see into. But since they're controlling both endpoints, all the other things like the Ethernet MAC address and the IMEI leakage, which had to be deliberate, can still be done even though you are doing HTTPS web browsing.

So my refrain on this is familiar to our listeners. Hotspot Shield has the right to do whatever they choose, but they cannot misrepresent what they're doing. And that's the CDT's only complaint. If users are truly informed about why their apps and services are free, and how Hotspot Shield is monetizing their use of the free service, then I and the CDT and the FTC would have no problem with that. But it's not okay to be offering a service which claims to provide absolute privacy enforcement while deliberately and by design violating that promise.

Leo: How would they, just out of curiosity, how do they know that information's being leaked?

Steve: I have no idea. This is just what I'm...

Leo: How would you know that? I guess you could test it by, I mean, there's no way they can monitor what Hotspot Shield is doing; can they? No, because it's on the Hotspot Shield servers.

Steve: They could go to www.target.com both with and without Hotspot Shield VPN and discover that they're getting different content, and they're seeing inserted content.

Leo: Well, you certainly could see that, yeah, yeah.

Steve: Yeah.

Leo: I guess that would be the only output.

Steve: Well, and they did reverse-engineer the app to actually see what it was doing.

Leo: Ah, okay, okay.

Steve: Yeah. So they took it apart. And again, this is why this has to be allowed. Researchers have to be able to verify these sorts of claims and then hold them accountable. And that's the only thing I want is that they say, okay, this is free, but here's what's going to happen.

Leo: Right.

Steve: Now, of course the people who are using a VPN, there are a range of reasons. And I could see that many people would be willing to trade a privacy disclosure that doesn't trouble them for the benefit of a VPN that does, for example, allow them to avoid local website filtering and so forth that they might be subjected to. Whereas others do care about having a VPN service provider who really honors their commitment, and probably for which they pay a fee, in order to, you know, because you have to have a business model that has this make sense.

Leo: Yeah.

Steve: And I did, I went to iOS to the App Store. And sure enough, Hotspot Shield VPN, free download, and you get what you pay for.

Leo: Yeah.

Steve: I got a kick out of - this is one little bit of errata from last week, Leo. We were talking about "Colossus: The Forbin Project." And I was saying, yeah, I thought that the actor looked familiar to me. I thought he was a Bond actor. Anyway, Ed Moreau sent me a tweet saying: "'Not famous' 'Bond actor' in 'Colossus' has starred in 'The Young and the Restless...'"

Leo: Yeah, he's a soap opera guy.

Steve: "...soap opera for decades." So, yes.

Leo: We now know Steve's secret viewing habits.

Steve: No, actually, I have dated women that have been into the daytime soaps. You know, what was it, "All My Children."

Leo: He looks familiar to me, too. I feel like he's been in something.

Steve: He does look familiar.

Leo: But I just - I can't place it. By the way, Alex Gumpel gave me his - he had the DVD, so he gave me a rip of the DVD so I can...

Steve: Oh, good.

Leo: Because you can't buy it. You can't get it.

Steve: That's gone?

Leo: Pardon me?

Steve: I mean, it's not available.

Leo: The DVD might be, but it's not streaming anywhere. It's not on Netflix or anywhere else. And I suspect the DVD even would be hard to get. This is a sad, actually, commentary on the state of the situation in the digital age. There's no reason why every movie ever made shouldn't be available to stream or download. But the movie industries are not bothering to digitize many great movies. And so you can buy it on a DVD in many cases, but that's not going to do you any good unless you have a DVD player, and fewer and fewer people do.

Steve: Yeah.

Leo: It's an opportunity missed. Why isn't any - you don't have to give it away, just why isn't it free to buy? Why can't you buy it on iTunes?

Steve: Right, right, because there's - with storage as inexpensive as it is now, it costs nothing to have it sitting there. And if a few people are going to - and, I mean, imagine all the people, after listening to us talk about it last week, who would have wanted to watch it for a buck if it was available.

Leo: Right. What's going to happen is it's all going to end up in lower quality copies on YouTube, you know, pirated, essentially. And that...

Steve: But that gets yanked down; right?

Leo: Well, it turns out "Colossus" is up there. It's not a very good quality rip.

Steve: Good [clearing throat].

Leo: Yeah [clearing throat].

Steve: So one piece of miscellany. Just yesterday I received email from Amazon telling me that the final book of the trilogy of the Rho Agenda was released. The book is titled "The Meridian Ascent." And so just to remind people, a lot of our listeners really enjoyed the original Rho Agenda trilogy, which was those teenagers running around New Mexico, finding an alien starship and getting up to some mischief. Then there was a prequel trilogy that told the story of Jack and Janet and their adventures, which predate their involvement with the teenagers in the middle trilogy. And now we have then the third sequel trilogy, which is the assimilation trilogy.

I've read, well, I've read them all. And I've read the assimilation trilogy twice because, when the second one came out, I wanted to reread the first one to remember it all. And I will read "The Meridian Ascent" when I finish my reread of the 19-book Frontiers Saga. Now I've just started on book 14. And it was 18 books, but the other one, the last one, an additional one came out while I was doing my reread. So I'm closing in on having read it twice because it's so good. And I get constant feedback from our listeners saying, oh, my god, this Frontiers Saga is the best thing I've read in years. It's like, yeah, it is.

And speaking of the best thing in years, I have a piece of email from Darren in Brisbane, Australia, from the middle of last month, saying: "SpinRite SMART data and relative drive performance," which was really interesting. And there's an interesting takeaway from this that I don't think I've ever talked about before. He said: "Hey, Steve." And I rewrote a little bit of this to paraphrase it for clarity. He said: "I've been using SpinRite on my Seagate NAS drives for years, each month or so picking the next drive" - I guess in rotation - "to check. A few months ago I had a couple of drives give up completely, so I replaced them with Western Digital Red drives."

He said: "I've been keeping track of the ECC corrected and seek error counts" - that is, those are what SpinRite reports while it's running on the drive. He said: "And I've noticed a correlation between SpinRite's reported ECC counts and the time required to perform a full Level 4 SpinRite pass. The more errors, the longer SpinRite takes." He said: "The slowest remaining Seagate drive reports" - let's see, wow - "reports 1,350 million ECC errors." So that's 1.35 billion ECC errors.

"The next faster drive, which is still slow, shows around 850 million. But the new WD drives report zero ECC and seek errors, and they also run MUCH faster under SpinRite. So," he asks, "does this difference in ECC counts indicate that the WD drives' overall performance will be much better than the Seagate drives?" And he says: "I will also be very curious how these drives perform with the new version of SpinRite, so I'll join your newsgroup when you get SQRL completed and are back to SpinRite development. Love the podcast." He says: "Have been listening during my commute to work for years. Keep up the great work."

So Darren, since you're listening, hi and thank you. And, okay. So, yes, drives have a nonzero temporal overhead when they encounter the need to correct. The good news is they can correct. The bad news is it kicks them out of the flow. Essentially, drives are hoping for non-error reads, which allows them to maintain their 1:1 interleave, that is, allows them to just move through sectors at full speed. But if they hit a sector that requires a lot of math, that blows a rotation. So you will lose at least a rotation while the drive stops and needs to perform CPU-intensive math. I mean, there's a lot of hardware assist, but again it knocks you - if you lose the ability to start reading that next sector, you've got to wait for the drive to spin all the way around again.

So there is definitely a performance hit from the need for ECC and/or reseeking. A seek error is where the drive just goes to where it believes a sector is going to be, waits for the first sector header to come along which identifies it uniquely, and then goes, ooh, crap, I'm on the wrong track. And then it has to move to a track on either side, if it just landed in the wrong place. So all of those things slow it down.

Now, the problem is SMART data is not guaranteed to be provided. The main parameters that SpinRite shows on those bar graphs, where we showed a picture a few weeks ago where one of the health parameters had been pushed down about halfway. That's 100% assured to have meaning. What SpinRite is doing is interpreting other data which the drive sort of leaks. And that's where SpinRite is able to get something like the ECC corrected rate and total, which no other utility does. But because it's not part of the formal SMART spec, we can't rely on it. So if it's there, we interpret it and show it. But it doesn't have to be there.

So the fact that the WD drive shows zero ECC and seek errors may well be that its design doesn't publish that leakage information which SpinRite picks up and reports, if it's available. But what you can count on is speed. So the fact that those WD drives are completing a SpinRite Level 4 pass in a fraction of the time relative to their size of the older Seagate drives, that strongly indicates that in fact the WD drives are not encountering nearly the number of error correction or seek errors that the older Seagates are.

So for what it's worth, if you're in a situation where you do have a bunch of drives, make note of SpinRite's given level total execution time because that's a very accurate measure of what's going on in the drive. The lower that is, the better, even if the drive isn't exposing SMART data, which gives you another dimension of numeric value to lock onto. So, neat question, and one that even now, after 12 years, we've never had before.

Okay. So a bit of closing-the-loop feedback with our listeners. Chris Beattie, whose handle is a contraction of "jabberwocky," says: "Why are enterprises fussing" - he's referring to that TLS v1.3 perfect forward secrecy stuff we talked about a couple weeks ago. "Why are enterprises fussing over TLS v1.3 perfect forward secrecy when they're surely doing man in the middle for internal traffic inspection anyway? What am I missing?"

Well, Chris, I wanted to clarify this for you and others. It's data centers who are concerned, not enterprises. So I wanted to just make sure that I made that clear. It's data centers whose internal networking monitoring topology they claim requires that they be able to see into the traffic transiting through their center. And the counter argument is, well, but if it's going, if it's passing through you, then tough luck because there's nothing you can do. But if it's terminating in a server in your data center, then it's decrypted at that server. And so, again, seems like a gray area. But anyway, it is data centers that are saying the nature of their networking needs are such that they have to be able to decrypt. Wow. But not enterprises. As we know, enterprises, as Chris says correctly, are probably already doing that with their own middleboxes.

Craig Naples says: "If we aren't mainly" - oh. He says: "If we are mainly relying on ISPs to block ports exploited in SMB attacks, would using VPNs to bypass ISPs leave us more exposed?" And that's a great point. So as I mentioned, most ISPs are blocking those ports - 137, 138, 139, and 445 - which are what SMB has historically used. Some have been dropping their filtering and allowing that traffic through, so we can't count on it. But as Craig notes, a VPN which bypasses all the things the ISP does would remove those blocks because our traffic is no longer subjected to them.

However, the good news is the VPN, because it looks like a network adapter, should be on the outside of our system's firewall. That tunnel, the VPN tunnel would be also transiting the NAT router. So it would lose the NAT router's protection, which would normally be our second line of defense after the ISP. But because the VPN is typically implemented as a pseudo network adapter, it's probably on the outside of our system's firewall. But it is worth checking.

One thing you can instantly do is fire up your VPN and go to ShieldsUP!. You should be, let's see - oh, no. That won't work, sorry, because we will be checking the VPN endpoint, the server that your traffic is emerging from, not the server at the end of the VPN tunnel. That's actually a common source of confusion for people who do use ShieldsUP! and wonder why they're seeing all kinds of open ports. It's because ShieldsUP! looks at the IP from which the traffic is emerging.

So anyway, I don't think, without - you'd have to have someone probe your port - I'll have to think about that. I didn't think this through all the way, whether unsolicited public traffic could get back to you. Depends upon what the VPN server's doing at its end. You're probably safe. And you also have the firewall, the local system firewall to protect you, as well.

And in a very nice piece of news Anthony Headley sent, he said: "From the latest SN, you should probably switch to this command-line interface" - which I didn't know of - "aws.amazon.com/cli." And it looks wonderful. That page is titled "AWS Command Line Interface." I didn't know there was one. It describes itself: "The AWS Command Line Interface is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts. The AWS CLI introduces a new set of simple file commands for efficient file transfers to and from Amazon S3 and runs under Windows, Mac, and Linux." So aws.amazon.com/cli. Thank you, Anthony, and for anyone who's interested.

Oh, and I also did, I was curious, I got my most recent S3 bill, and it was less than it had been. Dropped down to $2.08. And so I thought, okay, that's enough. I've got to find out how much data I have stored there because maybe it's not that much. So while I was putting the show together last night, I went to AWS, and they had something called the "AWS Cost Explorer." And so I said, oh, show me. And it said it'll take 24 hours for us to generate your - to, like, to populate your report from your request. So it hasn't been 24 hours. I just checked a few minutes ago, and it still says the same thing. So next week I will be able to say, okay, I am paying two bucks a month, and this is how much data I have up there. I have no idea how much, but I'm going to find out. So I'll be able to give people some sense for that.

Seth Meister tweeted: "Love SpinRite. Is there an easy way to run it on an external hard disk drive while actually using the Windows computer it's plugged into? Thanks." Okay. Now, the question, I guess, is easy. It's definitely possible, and many people do. If you fire up a VM, a virtual machine, and give it exclusive access to that external drive, then it looks like a physical drive, and SpinRite will run happily.

And this is actually far more practical than it might seem at first, since SpinRite and DOS is all you need, and they can run happily in a VM with 640K, yes, K, 640K, two thirds of a meg of RAM. And it will use virtually zero of your hosting machine's CPU. We've seen and shared photos on this podcast of people running even multiple VMs with multiple drives all running at once while they're using their computer. So it's not a mode that we formally support at this time. It's certainly something I'm going to look into for v7, once all of the 6.x technology is developed. But in the meantime, SpinRite 6 can run in a very tiny VM and allow you to use your computer while it's working in the background. It works perfectly.

Rene, looks like Feliu, he said: "If ransomware attacks a system, does it upload your data? If I get attacked, should I assume my files could be copied to the bad guys?" Okay. So Rene, the lesson here, and we've talked about this before, the rule of thumb is, if any malware gets into your system, you no longer know anything about it. That is, we know from rootkits that they're able to hide. We know that malware removers remove malware which then returns mysteriously.

So the problem is, I mean, the strict purist answer is, if a system is ever infected with something malicious, you can never absolutely trust it again. No one can tell you how little you should trust it, except you can't absolutely trust it because, you know, the only thing you could do would be to restore from a pre-infection backup. If you have a restore point and the malware didn't mess with that, you could restore from an earlier restore point. But you just - you're never going to know again after that has happened.

So classic ransomware does not upload. It merely encrypts. So that's the deal. It's just encrypting your data. If everything goes well, and you decide you need it back because you didn't have a backup, you pay them the ransom, and then they give you a key that lets you decrypt it. But still, somebody's been messing with your data. So you have to make a judgment on how important your data is, how crucial the privacy is, and how much you can trust that the malware didn't do anything else.

And I should note we have discussed malware, ransomware, that does more, that is also a trojan, that also leaves stuff behind, that also opens a backdoor. So ransomware is no longer just ransomware. And so that further strengthens the theoretical argument that, once your system's been compromised, anything could be in it.

Gary S. Martin asked of my discussion of S3 storage: "What encryption solution are you using?" He said: "You mentioned Duplicati in 2012 and Boxcryptor in 2014. Either of these?" And I should say, I'll take this opportunity to mention that I've switched to CloudBerry. CloudBerry Lab are the guys. I've discussed them before on this podcast, and in fact they quote my discussion of them on this podcast on their site. They did the encryption right. They offer a free version. They offer, for $30, a one-time pay for the software that does 256-bit encryption and Trust No One operation. It is massively cross-cloud platform. I mean, the list is just huge. They've got, like, the most popular, the also, and then everybody else.

I mean, and I should also mention they're what I use at GRC. It is CloudBerry Lab. I mean, they have ranges of plans from the individual to enterprise, and that's what I use for securely backing up GRC's servers at Level 3. So they're the people. As I was referring to earlier, I don't like this paying for no benefit for a software that I could purchase. Like you, Leo, I immediately purchase lifetime subscriptions for my TiVos because I love my TiVos, and I want to use them forever. That's just me. But Boxcryptor used to have a legacy version which you could still purchase, while they had the subscription plan. Now the legacy is no more. So it's like, okay, fine. CloudBerry is now my company. So that's the one I'm using.

Leo: Good to know.

Steve: Jeremy Malone asked: "What we need for your barcoded voting paper is the 'turbo entabulator' so the results are even faster." And I'll just note, I thought of that because somebody made a very good point, and unfortunately I didn't catch his Twitter handle go by. So anyway, he noted that human-readable barcode, or I'm sorry, human-unreadable barcode could also be hacked. So he asked the question: "Wouldn't a better solution be a human-readable paper output?"

And that's a very good point. I got a little carried away with my secret interchange language between the thing that runs the UI and generates paper, and then the turbo entabulator, which sucks that in. Whoever said that is 100% right. Better to produce a small paper readable summary in a format that an optical scanner can then read with high integrity, so that anybody can verify the paper output. There then the machine is just a UI to produce this high reading integrity, but still human-readable piece of paper. So great thought, Jeremy. And the person who - or a great question, Jeremy, about the entabulator, and also the person who said, hey, it ought to be human readable.

And lastly, Norbert Boron said...

Leo: What a great name, by the way.

Steve: Norbert Boron, yes.

Leo: I want that name.

Steve: He said, well, and this is why our listeners are so great: "If you rename Special Episode 85a to 85, and decrease all prior episodes..."

Leo: Yeah, not going to happen.

Steve: "...by one, then you'll have Episode 0." So very clever, Norbert. Thank you for that.

Leo: We had a couple places where we didn't - I forgot why that happened. Probably because we did something wrong in the feed, and earlier there was no way to fix a thing without pushing a different...

Steve: Either that, or I think we once did a special episode.

Leo: Maybe.

Steve: Of, like, something horrible happened, or wonderful, or something.

Leo: Oh, maybe that's what happened.

Steve: I think it was like a - we were like, oh, my god.

Leo: Like an extra episode.

Steve: Coming to you live because...

Leo: Yeah, it was, it was a special episode of Security Now! to warn and inform listeners of a serious zero-day bug in XP and Vista.

Steve: Ah. Now we do those every day. That was...

Leo: The animated cursor vulnerability. Dit dit dit, dit dit dit, dit dit dit. This just in. It was on April 2nd, which makes me a little nervous.

Steve: Oh, I'm sure I was skeptical even then. It was like, uh, okay.

Leo: It's only an 11-minute episode. So 11 minutes. Can you imagine?

Steve: No, I can't.

Leo: Most of our ads are that long now.

Steve: Yeah, that's time for me to take a sip of coffee.

Leo: Eleven minutes. We couldn't even get started in 11 minutes.

Steve: Ah, the good old days.

Leo: Ah, the good old days.

Steve: When a zero-day actually merited an emergency broadcast. Now it's like, yeah, we'll get to that next Tuesday. Yeah, we'll...

Leo: Little did we know. Little did we know where we were heading.

Steve: Yup.

Leo: That's fun. Well, I don't think we can easily renumber this without...

Steve: No, we're not.

Leo: We'd have to do a whole big Apache mod rewrite to say, well, if you ask for this, get that. And, oh, man, it would be...

Steve: No. But I appreciated Norbert's note that there was a buckle in the flow, and that we could remove that little kink and push all the earlier ones down. And then, with that, the Honey Monkeys would then be number one.

Leo: Be zero, Episode 0.

Steve: Be number zero, just belong.

Leo: Yeah, Honey Monkey Zero. It's where it belongs. Yeah, I think we'll keep it the way it is. I don't want to break all those scripts and so forth.

Steve: Just appreciated his observation.

Leo: Yeah. Yeah, we did, so we did an Episode March 29th, and then we burst in on April 2nd to bring you this breaking news.

Steve: Zero day, oh, my.

Leo: A flaw in Vista. Imagine that.

Steve: It's like the sky is falling.

Leo: We'll talk about "Game of Thrones" some other time, off the air, Steve Gibson.

Steve: Off the air. We're not into spoiling it for anybody; but, boy, I'm [glitch] twice now. They have been so good this year.

Leo: Our show is available for you to download. If you want to watch it live, you can. We do it every Tuesday, right after MacBreak Weekly, which is around 1:30, if you want to watch the live stream. It's really watching the live production of it. It's not a TV channel. You're not watching a live show, you're watching the production of it, 1:30 Pacific, 4:30 Eastern - I just don't want anybody's hopes to be all excited - 20:30 UTC. The live stream is at TWiT.tv/live on our website. You can also chat with us at irc.twit.tv. A lot of chatters during this show, talking about this stuff. It's great. This is, by the way, what did you say, our 13th year?

Steve: We're closing in on the end of 12.

Leo: Okay. So this isn't beginning the 13th yet.

Steve: Correct.

Leo: Soon. Next week?

Steve: I think the 17th or - I looked a couple weeks ago because I thought, I think it's around this time of the year it keeps happening. What was Honey Monkey's time?

Leo: I will tell you the date of Honey Monkeys. It was April, I'm sorry, August 18th, 2005.

Steve: Yup, 18th.

Leo: So on the 18th we enter our 13th year.

Steve: So not the next one - oh, wait, yeah. Next one will be the last episode of Year 12.

Leo: So this is the penultimate episode.

Steve: Yes, the penultimate episode.

Leo: Of Year 12.

Steve: Hang on for the ultimate.

Leo: Volume 12 is almost over. You can get the show in a variety of places. Start with Steve's site, GRC.com, because not only do you get the audio, it's the one and only place you can get the transcripts. They're nicely transcribed out. It takes a few days. But by the time they're up there, that's a very nice resource. It's also searchable.

Steve: It's usually Thursday morning Elaine mails the transcript, and then I get it posted.

Leo: And then he also has so much great stuff there, including his bread and butter, SpinRite, the world's best hard drive recovery and maintenance utility. But he also has lots of free stuff there. There's Steve's...

Steve: Everything else.

Leo: What was it, a mind ranging - what was that quote about Newton? I can't remember it exactly, but a free mind ranging widely. And that's him, right there, at GRC.com. That's the home...

Steve: [Crosstalk] that some of our listeners know.

Leo: Yes, they all know well. You also will find audio and video, we have video for some reason, on our site, TWiT.tv/sn.

Steve: With you talking like that, it sounds like it won't be forever.

Leo: Some reason, I don't know why. Well, people like to see it. And I think it gives them a little - not every episode. Every once in a while they watch the video just to make sure we haven't changed too much.

Steve: It makes it seem more real.

Leo: Yeah. It's more real.

Steve: They can check on the state of our hair.

Leo: Yeah, yeah.

Steve: Yours grew back.

Leo: Yeah, you shaved your head, and it never grew back. TWiT.tv/sn is where you'll find this show. But you know the best thing probably is to get one of those podcast programs, there are so many of them out there, and subscribe so you get the complete set. Wouldn't you like the complete set? Sure you would. Thanks for being here. We'll see you next Tuesday, barring a zero-day on Windows XP, on Security Now!.

Steve: Thanks, Leo.