A Tale of Two Counterfeits

Description: This week we, of course, start off by looking at what happened at Twitter last week. We look at Checkpoint's discovery of the headline-grabbing wormable DNS vulnerability that's been present in all Windows Servers for the past 17 years. We touch on last week's Patch Tuesday, Cloudflare's surprise outage, another glitch in Zoom's product, and seven "no-logging" VPN providers whose logs were all found online. We cover some other quick news and some interesting SpinRite development developments, then examine the problem of counterfeit networking equipment - which, as our Picture of the Week shows, is actually a big problem.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-776.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-776-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He's going to take a look at the Twitter hack of last week; the Cloudflare problem last week. We'll also check in with a couple of security researchers who made a little booboo reporting a flaw in a counterfeit device. And then Steve's going to delve deep into the problem of counterfeit network devices. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now!, Episode 776, recorded Tuesday, July 21st, 2020: A Tale of Two Counterfeits.

It's time for Security Now!, the show where we cover your security, your privacy, how the Internet works, and all sorts of interesting things because of this guy right here, the most interesting man in the world, Steve Gibson. Well, you kind of are. You don't drink beer, but you kind of - well, maybe you do. I don't know. He's the coffee version.

Steve Gibson: Yeah, I drink beer.

Leo: The coffee version of the most interesting man in the world, Steve Gibson of GRC.com. Hello, Steve.

Steve: We are closing in on the end of 15 years. We've got Security Now! Episode 776 for July 21st, titled "A Tale of Two Counterfeits." And I realized when I was done, maybe it should have been three. But I wasn't sure that just having two counterfeits would fit on the lower third because that was sort of a - that was pushing things to have the title that long. So I thought, okay, we'll just go with two.

Leo: Two's easier.

Steve: Yeah. But first, before we get into stuff, after last week's podcast, during which I rather clearly expressed my lack of understanding of some people's sensitivity to terminology, a number of our listeners reached out to share their distress and disappointment with me. I responded to each and every one of the people who wrote, and we engaged in very productive dialogue. I am embarrassed in the presence of their grace, and I sincerely apologize to those listeners whom I offended with my rant, which belittled something that they're sensitive to. That's not my place.

This is not to say that I fully appreciate the sensitivity to terminology that clearly has nothing to do with racial prejudice. But thanks to the conversations I had with some of our listeners, the way I can understand this is by seeing this change in terminology for what it represents. It's a proxy for some clearly very much needed progress toward the recognition of the full equality that we are each obviously entitled to by birth. So I get it. And the other thing I get is that no one is tuning in to this podcast for an update on my feelings about social justice. I am so impressed by the quality of our listeners. Twitter has probably never seen such civilized dialogue.

Leo: Yeah, it was really good, yeah.

Steve: I thank you all for putting up with me. Now let's talk about some stuff that I do understand.

Leo: Thank you for raising that issue. And I think no harm, no foul. I think one of the advantages of this terminology change, we've talked about this before, is it actually replaces things like blacklist and whitelist with deny list and allow list, which frankly I think are better terms anyway.

Steve: More accurate, yeah.

Leo: Much more. More communicative. More informational. And, yeah, if it hurts somebody's feelings, regardless of its genesis, then don't use it. It's okay. It's fine. And the trend continues. Apple has now joined that list. We were talking about the Linux kernel community last week. GitHub's doing it. So more and more we're getting rid of these terms that are just offensive to some. And I have no problem with it.

Steve: Well, and as I said, I really think what it represents is an awareness, an increase in sensitivity which has obviously been lacking for decades, I mean, like, you know, we're still there.

Leo: Yeah, and people, I understand with some merit, say, "Oh, come on, it doesn't have anything to do with that," or "That's a minor thing." But it's not for us to say. It's for the people for whom it's offensive to say, and I think it's appropriate if they're bothered by it to get rid of it. And there is a lot of unconscious racism in the way we speak. And if we can root that out, the better off we all are.

Steve: Absolutely.

Leo: Thank you, Steve. I appreciate that.

Steve: So this week, of course, as has probably every podcast you've done, Leo, we have to talk about what happened at Twitter. I mean, probably our listeners know, but it has to be covered because it was a major event for the industry.

Leo: Oh, yeah.

Steve: We're also going to look at Checkpoint's discovery of the headline-grabbing wormable DNS vulnerability that's been present in all Windows Servers for the past 17 years. We touch on last week's Patch Tuesday, look at Cloudflare's surprising outage and what was its cause, another glitch in Zoom's product, and seven "no-logging" VPN providers whose logs were all found online. We cover some other quick news; some interesting SpinRite development developments. Then we're going to examine the problem of counterfeit networking equipment, which as our Picture of the Week shows is actually a big problem because I can't tell which one is the fake. So I think another great podcast for our listeners.

Leo: You missed a good opportunity. Which one's the fake? That would have been a good title, too.

Steve: So our Picture of the Week is from a report that we'll be getting to at the end of the show that F-Secure did of their analysis of two fake Cisco routers. And this photo shows the authentic one next to the fake one.

Leo: Now, I haven't looked ahead. I think the left one is fake because the silkscreen is lighter on the label.

Steve: And you're correct.

Leo: But that was, I mean, look how close they look.

Steve: It could have been either. And in the text of their report they noted that the counterfeits used white ink for the port numbering, whereas the authentic Cisco used a gray ink. And you can see that it's a little bit brighter white on the left.

Leo: You can hardly tell, though. Wow.

Steve: And they said that the shape of the button was different. But, like, again, if you didn't have a real one, like, right next to it to compare with, you'd think the shape of the fake button was just fine because it's square and beveled and rounded just right. I mean, it's amazing how close these are. So anyway, I just thought it would be fun to make it

the Picture of the Week because we'll be talking about this later. But, boy, the issue of networking equipment counterfeits...

Leo: Amazing, yeah.

Steve: ...turns out to be a big deal. You know, it's not just Rolex watches and ladies' handbags that are being counterfeited. And we should just talk about Twitter. I think everybody by now, Leo, knows that there was a big event that happened on Wednesday of last week. The one thing that's missing from the most up-to-date information I've seen is any sense for who did it. And I'm sure that's, if they have any idea, that's embargoed because they're not wanting to get in the way of law enforcement or whatever forensics they may have.

Leo: Well, The New York Times fingered a guy named Kirk.

Steve: Oh. Hadn't heard that.

Leo: I mean, I don't think they've - that's the identity he assumed. They haven't associated that with a real person. But there is a great New York Times article where they talked - it's interesting because Kirk, before he perpetrated this, was reaching out to a bunch of SIM swappers. These are guys who aren't exactly malicious. They make money by hacking single-letter Twitter domains, or what they call "OG domains" - short, clever Twitter domains - and selling them to the highest bidder.

Steve: Like @6.

Leo: @6 or @y; right. But these guys - so Kirk was talking to them, saying look, I can do these things, you know, how much would you pay? But they got bored, apparently. According to their story, anyway. And they didn't follow up. But then shortly thereafter he did use his admin console to perpetrate this spammy hack.

Steve: So is he a Twitter employee, we think?

Leo: No.

Steve: Oh, okay.

Leo: Oh, you didn't - okay, got to read the story because what - and Twitter has not admitted to this. Whether it's true is unknown. But there seems to be good evidence that the way Kirk got this - Twitter only says "socially engineered our guys."

Steve: Right.

Leo: Is he got into their private Slack channel, where Twitter had pinned the credentials for the "god mode" interface. Wow.

Steve: That's a good one.

Leo: Oh, man. So all he really had to do was say, hey, let me into your Slack channel.

Steve: So it's like, do not write your password...

Leo: Yeah, it's like post-it notes.

Steve: Do not write your password on the blackboard.

Leo: Yeah, it's a post-it note.

Steve: When the camera crew is going to come in and do an interview.

Leo: Can you believe it? Now, that's the New York Times story, and I think that it's well sourced because they got screenshots from these guys, and they got a lot of stuff from these guys to verify that they were talking to this guy and so forth. But Twitter has not confirmed. So isn't that fascinating?

Steve: Well, I'm glad that we talked about this because you had more up-to-date information that I did. All I had was...

Leo: Oh, I loved this story.

Steve: ...yeah, Twitter's note from Friday. And, you know, they talked about 138 accounts; and, of the 138, 45 were messed with. And then eight people, they had some internal tool called Your Twitter Account or something that was used eight times. So, I mean, they're painting the best picture they can. It was funny, too, because I got an email from Jason in the afternoon on Wednesday saying, "Hey, Steve, we'd love to have you on Thursday morning to talk about this on our show. Can you do that?" And I said yeah, sure. So I attempted to tweet the news to my followers that I would be on on Thursday morning, and I was blocked. I got the weirdest, like, this looks like a bot is trying to post, so bot off. And it's like, what? I'm not a bot.

Leo: Yeah. They were suspending accounts, too. They were getting very aggressive about this, yeah. They didn't want anybody to talk about it.

Steve: Well, in fact because I'm sure yours was suspended, too, because you and I are both, what is it with the little...

Leo: Blue checks. We're verified.

Steve: Blue checked. Yeah, verified. And so they did a blanket block of all verified users. And then there were other people who had - anyone who had changed their password recently, that raised their flag of suspicion, so they blocked any posts from those accounts. I mean, they really did respond quickly, as quickly as they could.

Leo: Yeah. Yeah, this is the New York Times article. It came out on Friday. This is a fascinating story. The reason it's perhaps more important than just, oh, they hacked Twitter, which who cares, is that the President uses this, and many other leaders use this as a messaging system to tell the world what they're up to. It's terrifying to think that any of these accounts could have been compromised. And it was obvious when it started that it wasn't a password hack or your average everyday hack because accounts like Bill Gates's and Joe Biden's and Barack Obama's, all of which were hacked, almost certainly have higher levels of security on them.

Steve: Right.

Leo: They're not just using monkey123 as their password.

Steve: This clearly had to be...

Leo: It looks to me to be an inside job; right?

Steve: It was obviously an inside job, yeah.

Leo: Yeah. That was our immediate conclusion.

Steve: Well, and the nature of the tweets. It was a two-for-one deal on bitcoin. Whatever you send me, I'll double it and send it back to you. And the last I heard was, I think, well, I said on Thursday, on Jason's show, that I had seen $300,000 had been transferred into the bitcoin account. I haven't seen that again. I've seen more like 120,000. So some amount of money was made.

Leo: It's hard to tell. The nature - we know the bitcoin account, so you can look at the blockchain and see what was going on with it. But hackers routinely will put money into it, take money out, to obfuscate what they're making.

Steve: Ah.

Leo: So, like you, I saw 100,000. But the best information came from Binance, which is a bitcoin exchange, which said they had blocked a quarter of a million dollars of attempted transfers into that account. So that's their customers saying, "I'd like to give some money to Joe Biden and get it doubled." So it might have been a more successful hack than it appeared to be. And if it weren't for Binance blocking

those transfers it could have been a lot more money. It does make you wonder, though, if you had this god mode, what would be the best...

Steve: Yes. Why use it for this?

Leo: Yeah, what would you do with it? And maybe this is misdirection. Maybe Kirk is really, you know, Vladimir, and he's up to other things. So we don't - that's why I wonder what's really going on here.

Steve: Well, and what I said when I was talking to Jason on Thursday was that, you know, I don't know whether he knew, but you and I talked years ago, back at the Sony Entertainment hack, where there was an advanced persistent threat that had been found in their network. And I said famously on the podcast, "I don't want the job of trying to secure something like Sony."

Leo: Oh, yeah.

Steve: You know, that's impossible. It's, I mean, literally impossible. And so what this suggests is that clearly it was a mistake, if the credentials for this thing were stapled in the Slack channel so that it could be seen. But also the idea that the bar is not higher to doing something like that. And I talked about how there are protocols that use a majority voting approach with crypto, where compromising a single individual wouldn't render the company vulnerable.

A secret, a shared secret could be shared, for example, among nine people, but you've got to get three of them to all agree in order to cause something to happen. That way, if you have a large enough body of people who can be involved, you're not worried about getting locked out because Maury is gone for the day. But at the same time, I mean, it's protecting the people from being suckered into doing something with a social engineering attack. So I think this also suggests that Twitter has become, as you said, Leo, so crucial. I mean, like bizarrely important on the global...

Leo: Bizarrely important.

Steve: On the global scale.

Leo: Well, it also - it's kind of unconscionable because Twitter, it's not the first - so a contractor leaving his job at Twitter two years ago, you might remember this, disconnected President Trump's account, deleted it. And then of course Jack's account, the CEO, his account was hacked a couple of months ago. You would think that at this point Twitter would kind of be on notice that maybe they should be doing a better job.

Steve: Yeah, this is just not a toy anymore that allows people to send 140 characters between their phones. It's become a lot more than that. So anyway, hopefully...

Leo: And we should say, even though The New York Times says that it was the Slack channel, you know, they got that from hackers, and who knows if that's true.

Steve: Right, right, right.

Leo: But Twitter doesn't say.

Steve: So "allegedly" to everything so far, unless we get it officially. And who knows when and/or if what we'll get officially from Twitter.

Leo: I doubt we'll hear, yeah.

Steve: Yeah. That's what's known now. And again, sort of puts the industry on notice that this stuff, which just started off being kind of like, oh, look. Are they making any money yet, Leo?

Leo: No.

Steve: Or are they still waiting to make money in the future?

Leo: They've had profitable quarters, but they're not exactly rolling in the dough.

Steve: We don't really have an economic model, but we're going to let the whole world use this to talk to itself, and good luck. We'll make it up in volume.

So SIGRed. As Checkpoint Research said: "This is not just another vulnerability." This month's big, scary, wormable vulnerability turns out to have been present in Windows Server versions since Windows Server 2003, which actually did come out in 2003, unlike Windows 10 2004, which just came out in 2020. But anyway, we'll get to my Windows rant a little bit later.

This problem has been present in all subsequent versions of Windows Server since, including Server 2019, which is the most recent release of Windows Server. So without knowing it, we've been living with this in our midst for the past 17 years. Its discoverer was Checkpoint Research, as I mentioned, who named it SIGRed. And I'll explain where SIG - SIG as in signature because it's about DNSSEC signing stuff, or signing records.

It was assigned a CVE-2020-1350. And I'm always suspicious when I see such a low CVE number. I wonder if they're going to have to start randomizing them because you could tell how old it is from how small it is. We're in July of 2020. So 1350, that happened right near the beginning of the year. And it's like, uh, okay, especially considering how serious the guys at Checkpoint think this is.

So it's wormable, meaning that it can propagate among any and all Windows Servers that can be induced to make a DNS query. And it turns out there are lots of clever ways that can be done, and the Checkpoint Research guys did all that. It's triggered by the receipt of a specially crafted DNS response. And since Windows Server services runs with elevated system privilege, if it's exploited, an attacker gets full domain admin rights,

effectively compromising the entire corporate infrastructure. And many who looked at this realized this could have been a flash worm of the sort like Slammer, which remember it took, was it 30 minutes to take over all the vulnerable systems on the Internet. It just exploded.

Leo: Geez, Louise.

Steve: Because it was a self-propagating worm. So, yeah. And the way Checkpoint explained their discovery, that is, why they went looking was sort of interesting. They wrote: "Our main goal was to find a vulnerability that would let an attacker compromise a Windows domain environment, preferably unauthenticated." They said: "There's a lot of research by various independent security researchers, as well as those sponsored by nation-states. Most of the published and publicly available materials and exploits focus on Microsoft's implementation of" - and no one's going to be surprised by this - "SMB (Server Message Blocks), i.e., EternalBlue, and RDP (Remote Desktop Protocol) BlueKeep protocols, as these targets affect both servers and endpoints."

They said: "To obtain domain admin privileges, a straightforward approach is to directly exploit the domain controller. Therefore, we decided to focus our research on a less publicly explored attack surface that exists primarily on Windows Server and domain controllers: WinDNS." For anyone who's interested in their really detailed tech stuff, I've got a link in the show notes because it's very detailed and, well, frankly, and it's wonderful, and takes us step by step through Checkpoint's process. So I'll just hit the high points.

For every query type that a DNS server makes, there is a corresponding reply. What Checkpoint found was a classic type conversion flaw, a math result variable sizing mistake in the parsing logic for the reply to a SIG, as in signature record, which is part of DNSSEC. The extension's, you know, DNS Security for DNS. They discovered literally by reverse-engineering I think it was dns.exe, which is the service that does WinDNS. They studied the code, the reverse-engineered code, and they found a mishandling of values between the 16-bit fields which are used by the DNS protocol and the 64-bit register math used by the code's compiler.

All coders know that if a 64-bit value is calculated to allocate memory, or even 32, that is, larger than 16 bits, so 64-bit or 32-bit, calculated to allocate memory, and if the result is larger than 65535, which is the maximum absolute quantity that can be represented within 16 bits, then the least 16 bits of the larger value will be a small integer. Basically it's the amount of the overflow over 65535. And if that smaller integer 16-bit value was then used to allocate memory for a buffer, the resulting buffer will be much too small to hold the larger calculated amount of data. And of course that's exactly what happened. They discovered that - and I just hit the spacebar, so I lost my place in my notes.

Leo: I discovered a blank page.

Steve: Discovered, yes, something non sequitur - that by sending a DNS response containing a larger than 256K SIG record, they could cause a controlled heap-based buffer overflow of roughly 64K, meaning they had a lot of excess buffer to work with. This is not a few bites that they have to be clever about. And for hackers that's the golden keys to the server kingdom. They concluded their write-up by noting, they said: "This high-severity vulnerability was acknowledged by Microsoft and was assigned CVE-

2020-1350." And as I said, I didn't go to look at the date of it, but it had to have been a while ago.

They said - this is Checkpoint who found the problem. "We believe that the likelihood of this vulnerability being exploited is high, as we internally found all of the primitives required to exploit this bug. Due to time constraints, we did not continue to pursue the exploitation of the bug, which would include chaining together all of the exploitation primitives; but we do believe that a determined attacker would or will," they wrote, "be able to exploit it. Successful exploitation of this vulnerability would have a severe impact, as you can often find unpatched Windows domain environments, especially domain controllers. In addition, some Internet Service Providers may even have set up their public DNS servers as WinDNS.

"We strongly recommend users to patch their affected Windows DNS servers in order to prevent the exploitation of this vulnerability. As a temporary workaround, until the patch is applied, we suggest setting the maximum length of a DNS message over TCP" - because DNS over UDP is restricted to the size of a single UDP packet, so it can't do a 64K - "which," they said, "should eliminate the vulnerability. You can do so by executing the following commands."

And basically I have it in the show notes for anyone who's interested. It's just a registry command to add an HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\DNS\Parameters key. It turns out you can add a parameter, TcpReceivePacketSize, and you can set that to FF00 in hex, which is 256 bits shy of 64K. I mean, sorry, 256 bytes shy of 64K. And so that ends up having priority over DNS, and it would prevent the overflow. And also it's rare that you're going to have any valid DNS that's right up to that 64K limit. So it wouldn't break anything.

So anyway, this is all only now becoming public, aside from the fact that has apparently been known for quite a while, because Checkpoint disclosed this responsibly. They whispered into Microsoft's ear, apparently some time ago, and Microsoft's July Patch Tuesday, last Tuesday, fixed this bad boy. So as long as you're updated and patched from last week, you're okay. If not, you should do that. And if for some reason you can't, then these two commands, the second one reads "net stop DNS," followed by "net start DNS." So you need to add this registry key, or change its setting if it's already there, and then restart the DNS service if for some reason you can't update with last Tuesday's patches.

And speaking of last Tuesday's patches, this month we received updated code from Microsoft to remove 123 security flaws from 13 products. Happily, none of the security bugs fixed this month have been discovered being exploited in the real world, though a bunch have been shown to enable remote code execution. And Leo, when you scroll through the whole list, oh, my, I mean, the scroll bar shrinks down, and you think, what has happened? Although the most worrisome of all of those, and that's actually my cut down list...

Leo: This is the edited version.

Steve: ...of just 30 remote code execution vulnerabilities.

Leo: This is not good. Oh, man.

Steve: The actual list, I was scanning it. ZDNet posted it. And it was just like, oh, my goodness. So it's quite sobering. And we've seen the damage that can simply be done by elevation of privilege bugs. Since today's operating systems are hosting so much content from so many various sources, maintaining isolation and control among them is one of any modern system's top jobs. This month I didn't even attempt to count those problems that were fixed. But as we noted, I decided to parse the list for the even worse remote code execution vulnerabilities that were just eliminated last week.

There's .NET Framework Remote Code Execution Vulnerability. And I'm not going to read all these over because my mouth will run dry. We've got SharePoint Server, Windows Font Driver, Windows Font Library, Microsoft Graphics Components, Microsoft Graphics, three Jet Database Engines, Microsoft Outlook. PerformancePoint Services, whatever that is. Excel, Office, Project, two in Word, VBScript Remote Code Execution, Visual Studio. Address Book has a remote code execution vulnerability. Remote Desktop Client. Oh, who's surprised?

Another LNK, L-N-K, file remote code execution vulnerability. They've fixed that now, what, two months running. Windows DNS Server. Well, that's the one we just talked about. Visual Studio and Visual Studio Code. Well, they probably use the same code, so okay. Visual Studio Code ESLint Extension remote code execution vulnerability. And then six Hyper-V Remote FX vGPU remote code execution vulnerabilities.

You know, this is just a daunting list, I mean, when you step back and think about it, of serious vulnerabilities to be fixed. Every four weeks we get one of these. And we've been seeing this all year. This is not a brief anomaly. It's the way it is now. And on Windows Weekly I've been hearing Paul and Mary Jo lamenting what sure does appear to be the collapsing state of affairs with Windows. So I know I'm not an outlier here.

And just get a load of this bit of explanation from a page in the last week on BleepingComputer. Oh, they posted it on Friday. The headline read: "Microsoft fixed an issue where the Disk Cleanup maintenance utility could cause boot failures when launching automatically after installing Windows 10 version 2004 build 19041.21." BleepingComputer wrote: "To prevent this issue, Microsoft is using an automated troubleshooter instead of applying an update block to prevent Disk Cleanup from launching on its own and causing boot issues until the users install the Windows version 19041.84 update which comes with a fix for this bug.

"Microsoft says: 'This troubleshooter automatically runs twice. It runs for the first time on all devices on Windows version 19041.21. Then it runs again after devices are upgraded to Windows version 19041.84. This troubleshooter cannot be run manually.'" Then BleepingComputer said: "To see if the troubleshooter has launched on your device, you have to check recommended troubleshooting history by going to Start > Settings > Update & Security > Troubleshoot > View troubleshooting history." What?

So, wait. "To prevent this issue, Microsoft is using an automated troubleshooter instead of applying an update block to prevent Disk Cleanup from launching on its own and causing boot issues." What Windows has become would even confuse Rube Goldberg. And as a developer, I can totally sympathize with the impossible task Microsoft has undertaken.

Leo: A lot of legacy code.

Steve: Pick any one of those patched vulnerabilities last week. Microsoft Office, for example. It has a remote code execution vulnerability? Of course it does. And not only one. We'll almost certainly have another couple fixed next month, and some more the

month after. Office alone has grown so massive and so sprawling that it can't help but to have hundreds of still-unknown exploitable bugs.

How did this happen? It's really very simple. Microsoft decided quite rationally that we wanted features more than we wanted security. And I can't say they were wrong. Features are visible. Bugs are hidden. Ever purchase a used car with a fresh paint job? That's what Windows gets every few years. The same increasingly old and increasingly creaky operating system with a fresh paint job. Windows has become the used car of operating systems. But it's the one most of us are driving. Beep beep.

Leo: You should read Neal Stephenson's book "In the Beginning There Was the Command Line." He has a great automotive analogy for the operating systems. It's very funny. It's very well written. And yes, Windows is the one that's beep beep. They run it in the corner, yeah.

Steve: Well, it is what everyone, I mean, it's still far and away the majority OS.

Leo: Oh, yes. And your software has to work with it. But do you someday envision a time when you can retire and just never again boot into Windows?

Steve: Yes. I do. And in fact, one of the things that I've learned during the beginning of this round of SpinRite work is that I can't put off switching to UEFI booting for SpinRite.

Leo: No. That's right.

Steve: Well, UEFI booting for SpinRite means no DOS.

Leo: Right.

Steve: So I will be becoming native booting in the short term here.

Leo: Nice. Interesting.

Steve: I'm going to do that immediately after finishing 6.1 because I want to still be able to be used by all recent machines.

Leo: So no more FreeDOS.

Steve: Yeah, I do look at the day when it's like, oh. I mean, when I hear Paul and Mary Jo saying, you know how there are these systems that aren't yet qualified to run Windows 10 2004? And Mary Jo says, "That's a good thing. Don't ask for it."

Leo: No hurry. No hurry.

Steve: Believe me, you don't want that.

Leo: No rush. Well, also it's a feature update, with features no one cares about. So Microsoft could just relax a little and slow down.

Steve: And actually, you know, I guess in my analogy that would be touchup paint.

Leo: Yeah, yeah. Bondo.

Steve: You know? No one really notices those little scratches.

Leo: It's just Bondo.

Steve: So Cloudflare had a big embarrassing outage. And it was one single measly little line in a router's config file. And suddenly Riot, GitLab, Patreon, Authy, Medium, Digital Ocean, and countless others - including, somewhat ironically, Downdetector - became unreachable and dropped off the 'Net.

Leo: Isn't it ironic that Downdetector was down?

Steve: Downdetector went down, yeah. Not their fault, of course.

Leo: No. Well, it was their fault. Not Downdetector's fault.

Steve: No, but I mean...

Leo: Cloudflare's fault. Right.

Steve: Right, right, right. So last Friday evening, while Twitter was updating the world about its massive hack, John Graham Cumming, our friend, was describing what happened at Cloudflare. I have the link to his more detailed post in the show notes. But basically I clipped out two pieces. He said: "Today, a configuration error in our backbone network caused an outage for Internet properties and Cloudflare services that lasted 27 minutes. We saw traffic drop by about 50% across our network.

"Because of the architecture of our backbone, this outage didn't affect the entire Cloudflare network and was localized to certain geographies. The outage occurred because, while working on an unrelated issue with a segment of the backbone from Newark to Chicago, our network engineering team updated the configuration on a router in Atlanta to alleviate congestion.

"This configuration contained an error that caused all traffic across our backbone to be sent to Atlanta. This quickly" - and I was like, whoops. "This quickly overwhelmed the Atlanta router and caused Cloudflare network locations connected to the backbone to fail. The affected locations were San Jose; Dallas; Seattle; Los Angeles; Chicago;

Washington, DC; Richmond; Newark; Atlanta; London; Amsterdam; Frankfurt; Paris; Stockholm; Moscow; St. Petersburg; So Paulo; Curitiba" - is that a place?

Leo: Yeah, Curitiba, yeah.

Steve: Okay, "...and Porto Alegre. Other locations continued to operate normally." What other locations? Are there any others?

Leo: Oh, yeah. You saw this beautiful graphic. This is gorgeous. I think this says it all. That hotspot is Atlanta. The white spots where there are no traffic, those are the NOCs that are down, right there.

Steve: Right.

Leo: Boom.

Steve: He says: "This was not caused by an attack" - of course speculation went wild as soon as Cloudflare became inaccessible. "This was not caused by an attack" - oh, and BleepingComputer was also taken down - "...not caused by an attack or a breach of any kind." He said: "We're sorry for this outage and have already made a global change to the backbone configuration that will prevent it from being able to occur again."

Leo: Oh, good.

Steve: But in the posting he did offer some interesting technical detail about their setup that I thought our listeners would find interesting. He said: "Cloudflare operates a backbone between many of our datacenters around the world. The backbone is a series of private lines between our datacenters that we use for faster and more reliable paths between them. These links allow us to carry traffic between different datacenters without going over the public Internet."

So what we used to call a "leased line," you know, where you're actually buying, probably, well, I'm sure fiber optic, some fiber optic carriage owned by AT&T or someone, just raw transit. And there's no routers in line. It's just these two points, Point A and Point B, are interconnected. And in fact it's probably a big honeycomb with direct links to all of the various major datacenters.

So he said: "We use this, for example, to reach a website origin server sitting in New York, carrying requests over our private backbone to both San Jose, California, as far as Frankfurt or So Paulo. This additional option to avoid the public Internet allows a higher quality of service, as the private network can be used to avoid Internet congestion points. With the backbone, we have far greater control over where and how to route Internet requests and traffic than the public Internet provides."

And of course we recently talked about BGP routing mistakes. They're easy to make. And when a mistake is made by an organization the size of Cloudflare, they're also hard to miss. And this was exactly that. There was some backbone congestion in their Atlanta, Georgia datacenter. So the team decided to remove some of Atlanta's traffic by rerouting it to other datacenters on the backbone, essentially removing some routes that were

pointing to Atlanta, where Atlanta's BGP was advertising a bunch of routes. They thought, okay, we're going to reduce its advertisement spread so that other routers on the backbone will pick it up.

But there was an unappreciated comparison in routing preference levels which resulted in this unanticipated result. Instead of removing the Atlanta routes from the backbone, the mistake caused the Atlanta router to start leaking all BGP routes into the backbone. So Atlanta, remember we've talked about this before, the vocabulary of BGP in routing is weird. So they said Atlanta was inadvertently advertising itself as the proper destination for all of Cloudflare's backbone traffic. The other routers at the other datacenters that received Atlanta's updated BGP table, which essentially amounted to a bold "come hither," they shrugged and said okay, and Atlanta was immediately buried and collapsed. As John put it: "With the routes sent out, Atlanta started attracting traffic from across the backbone."

So, yup, they were embarrassed and apologetic. And as we said, they already put safeguards in place so that nothing like that can happen again. Let's hope that Twitter is going to do the same thing.

Leo: Probably Twitter doesn't run BGP anywhere important.

Steve: No. And you could just imagine. I mean, I'm sure the team really knows its stuff. This is the kind of thing where just it was a 100 versus a 200 in the priority list. And it wasn't until it happened that someone said, uh, what's happening? And then they had to dig down and find it. The reason it took 27 minutes was that it wasn't obvious what the problem was.

Leo: Right.

Steve: And so as soon as they found it, it was like, ah, and then they fixed it, and then the traffic got itself cleared up.

Leo: I have to say I'm impressed by the forthrightness that Cloud - I mean, this is the code. This is the error in the code.

Steve: Yes, exactly. They're actually showing the line, yes.

Leo: Yeah. And I think that's great. That's always reassuring.

Steve: That's why we love them.

Leo: Yeah, it's why we love them. And we would love John because he's a real geek, and he understands. He gets it, yeah.

Steve: Yeah. Any outfit that puts up a wall of lava lamps and aims a video camera at them as their source of entropy...

Leo: And chaos.

Steve: What's not to love?

Leo: Yeah, yeah. They're cool. They're really cool.

Steve: So Zoom had a Vanity URL flaw. That's literally what they call it. They recently repaired another glitch which was discovered by, not surprisingly, Checkpoint. Those guys have been busy lately. Checkpoint responsibly and privately reported it to Zoom, and it was quickly fixed. After it was fixed, the world learned of what had once been a problem. I phrase it this way because I continue to see that the tech press appears to have fallen in love with the term "zero-day."

Now, everything is a zero-day vulnerability, even when it's not. This incident was widely being called a "zero-day vulnerability." It never was. Those researchers we talked about last week, those two who disclosed severe vulnerabilities in C-Data's equipment, they created multiple zero-day vulnerabilities by irresponsibly going public with their disclosure and not giving the manufacturer any opportunity to respond, to fix the trouble, and push patches.

But Checkpoint and Zoom working behind the scenes to fix a problem before it became publicly known is not a zero-day. Vulnerabilities are not zero-days when they are fixed before they are publicly revealed. So I'm hoping that the tech press hears this because it's a catchy term. But if we start labeling everything a zero-day, then that's crying wolf. And when something actually is one, then no one's going to notice. So I hope we keep our terms straight.

Anyway, it turns out that Zoom allows the use of so-called "Vanity URLs." And that's what they actually call them. The link, I have it in the show notes, is "Guidelines for Vanity URL Requests." You need to register your intention to basically create a subdomain of Zoom.us. So Checkpoint discovered that due to improper account validation, any meeting ID could have been launched using any organization's Vanity URL, even if a meeting was set up by a separate individual account which had no relationship to the organization. It's unclear from Checkpoint's disclosure whether there was actually any subdomain validation, though I assume there must have been some. I mean, I hope so.

So what Checkpoint wrote to explain this is, they said: "Upon setting up a meeting, an attacker could change the invitation link URL to include any registered subdomain." Which is why it doesn't sound like there was much validation. For instance, they said, if the original invitation link was https://zoom.us, right, regular, and then /j/ and then the serial number for the session, they write, the attacker could change it to https://organization's name, you know, like IBM or anything, Mozilla, .zoom.us/j/ and then the serial number.

So a victim receiving such an invitation would have had no way of knowing the invitation did not actually come from the actual organization, ibm.zoom.us or mozilla dot. So, you know, just a phishing/spoofing attack. But wow. That shouldn't be allowed to just be done. So as I said, it's unclear whether it was that easy. I hope it wasn't. And since it's fixed, it didn't merit any further digging on my part. I didn't care to go any further.

But I noted that Checkpoint added an interesting statistic factoid at the end of their disclosure write-up. They wrote: "It's worth noting that 90% [nine zero percent] of cyberattacks today start with a phishing email." They said: "To make sure you're doing

enough to protect your organization's attack vectors, we suggest that you read the whitepaper 'Humans are Your Weakest Link' to discover the daily risk posed by phishing emails." If anyone's interested, you could probably google "phishing attacks put your business at risk." That's in the URL. Or I have the URL in the show notes. If you click it, they solicit your name and organization name and email address. But maybe it's worth it to you. I mean, Checkpoint. So it's not like nobody.

And I was curious also just sort of to see where Zoom was in the ecosystem world. They're currently at 35.87% of the global web-based tele-whatever they are, teleconferencing solution, 35.87%; GoToWebinar, 22.44; and Cisco Webex at 17.18. So they're the biggest now, but they're not bigger than everybody else combined or anything like that. So still worth, you know, they're in the spotlight because they're catching all the headlines, and everybody is talking about "Zooming" now apparently has become a thing. Turns out, and one of our sponsors will be glad to hear this, not all VPNs are created equal.

Leo: Oh, boy, I saw this one, yeah.

Steve: We learned this week some specific details of something we probably always suspected, which is it really does matter which VPN provider one chooses. There's a site, VPNmentor, which obtains its revenue from affiliate links to well-known and upstanding VPN providers. One of them is a current sponsor on the TWiT network. Recently, the user connection and activity logs of seven, on the surface apparently different, free VPN providers who all boasted about their "zero logging" services, were discovered on the Internet. That is, yes, the connection and activity logs of seven VPN providers that don't log were discovered on the Internet, in the cloud, on an Elasticsearch database instance.

VPNmentor's research team, led by Noam Roten, discovered the database containing a staggering one billion database entries associated with approximately 20 million users. So a little bit of division, an average of 50 log entries per user, despite the fact that each of the VPN services advertises, as I said, they are no-log VPNs. The database contained Internet activity logs with personally identifiable data and email addresses, cleartext passwords, IP addresses, home addresses, phone numbers, device IDs, and other technical details.

The good news is they're not mainstream VPNs. They are UFO VPN, Fast VPN - because I don't think Slow VPN would get many takers. So Fast VPN, Free VPN, Super VPN, Flash VPN, Secure VPN because why not, and Rabbit VPN because rabbits are fast. So while they all have separate names, they all appear to be connected to a common app developer who apparently white-labeled one product to multiple companies who were not very imaginative about their names. But from the outside, the typical consumer would have absolutely no way of knowing.

I put one of the ad screens from Super VPN on the show notes. And it says, you know, it's got Home; the Privacy Policy, that's where the no logging is said; Terms of Service. And they say: "Best VPN solution for Android. Unblock the world freely and easily. Reclaim your right to privacy. Enjoy the open Internet." And then there's the you can download their app for iOS from the App Store. Get it on Google Play. I mean, it looks absolutely like you'd expect. And there's a big checkbox on the app showing, yes, you're checked. And so for Super VPN, when you go look at the details, it's the best unlimited VPN proxy for Android. Google Play Store has 4.6 stars and more than a million downloads. One of the other ones I noted was 10 million, more than 10 million downloads. The App Store ranks it at 4.9 stars. The developer is Nownetmobi in Hong Kong.

So any unwitting user might be forgiven for thinking, hey, looks great. A free VPN service sounds perfect. Of course we know today a VPN service need not be expensive. But if actual privacy and security is one's goal, I'd rather pay a fair price and have that actually provided. It clearly does matter which provider one chooses. And if all of your Internet traffic is coming out of a sketchy provider in Hong Kong, that just happens to be logging all of the data they have about you onto an exposed Elastic Cloud instance on the Internet, that's not the VPN provider that you want to use. Just saying.

Apple recently updated its iOS and macOS with a handful of useful security patches. There's not much detail because Apple doesn't provide much. But I scanned them, and they looked important. Useful, at least. So for me, this happened Friday, I think, and I had to go into my general settings under both iPhones. My iPhone 6 couldn't come to 13. It came to, I don't know, 12.4.8 or something. My iPhone 10 or X did go to 13 point whatever. So anyway, I mentioned it in case you have to sort of solicit that update from Apple, as I did.

Leo: You don't, normally. The way it works is yes, you do, and I did, to get 13.6. But the reason is because they do staged rollouts. So everybody will get pushed it eventually, and you'll get a notification that there's an update. But they don't do it right away. So, yeah, once you read that there's an update, you can absolutely go to the updates and get it. But they typically won't push it to you for a week or two after that. Then they'll say, hey, there's an update. Which actually is kind of the right way to do it because it gives people who want it right away a chance to get it, but it also gives it some time to sit and stew in case there's any issues.

Steve: Yeah. And how many times have - yeah. How many times have we said, agh, can I uninstall this thing?

Leo: Yeah.

Steve: I also noted that Firefox Send is still not receiving. And this is becoming a curious outage because you wouldn't think that requiring everyone to have an account tied to a verified email address, when that was already an option, nor adding a Report Abuse button would be a heavy lift for Mozilla. So it's beginning to feel as though perhaps something more substantial might be going on behind the scenes. And if they're able to make Firefox Send better by making it even more resistant to abuse in other ways, so that it doesn't again immediately fall victim to malware purveyors, I would say it's probably worth waiting for. So I've just sort of been checking back. And it's like, it's been a while now. It's like, ah, that's sort of interesting that it's still off the grid.

I have a nice piece of listener feedback from Joe Lyon. I saw his tweet a few days ago. He said: "I just had SpinRite save my Dell E7450 Win7 Pro laptop." He said: "I recently booted the machine, and all my desktop icons, SQRL login, et cetera, were gone. I did a search online, and it seems to be related to a corrupted profile in Windows. There were detailed steps to take to recover my profile and try to get all of the information back. Before taking all those steps and doing all that work, I thought I'd try SpinRite. I booted the machine with FreeDOS and ran SpinRite on Level 2." Actually, that machine has a solid-state drive, so that's what you want is just do a Level 2 scan. "After about an hour I rebooted into Windows and, voila, it booted properly; and all my files, icons, programs were back where they should be. Thank you, SpinRite and Steve."

Which leads me into a little bit of update on where SpinRite is. And frankly, I have so much news on the SpinRite R&D front that I hardly know where to begin. For one thing,

our work so far has suggested and is suggesting to me that there may be far more that can be done with solid-state storage than I was expecting. The timing results we're seeing from the benchmarks suggest that there's far more going on under the hood than we might expect from solid-state storage. But then, you know, when thinking about it, of course the SSD engineers would be squeezing every last possible bit, literally, out of the technology that they're able to, just as was done with hard drives, which is what resulted in such a large recovery margin.

So I'm beginning to better understand why SpinRite has had so many reports of success with the recovery of data from solid-state storage. I'm beginning to think that we'll eventually have some sort of solid-state storage assessment tool unlike anything that's been done before. So that's all I'm going to say at this point.

Leo: Oh, you're cagey. Come on.

Steve: My spidey senses are kind of saying, oh, that's interesting.

Leo: Interesting.

Steve: In other news, there's a timesaving approach that SpinRite will be able to use for any mass storage media, spinning or solid-state, when the actual transfer rate from the storage medium exceeds the transfer rate of its link to the system. That won't be an issue with NVMe storage because they've solved the link problem. There's no serial SATA bus there. But it will when a fast-spinning drive capable of more than, for example, 300 MB/sec is stuck behind a SATA II link that maxes out at 300 MB/sec; or with a fast SSD that can read faster than SATA III's maximum, which is 600 MB/sec.

It's possible to ask a drive to verify that it's able to read from its physical media without actually bothering to transfer the data across the serial SATA link. We have found that some drives apparently cheat when asked that, and immediately say, yeah, no problem. So SpinRite will be carefully testing drives first by deliberately inducing an error, then checking to see whether the drive tells the truth and says whoops, or lies when asked and says no problem. And the reason this is so exciting, when it can be done - and, I mean, basically drives are cheating. They're in breach of spec when they just blow off what's called the "read verify" command. But we've determined that a bunch of people have them that do.

So SpinRite will verify that it's authentic. When it can be done, for drives that faithfully honor this read verify command, SpinRite will potentially be able to perform a full media test at a very practical speed. One of the people testing the R&D code has a 500GB Samsung 850 EVO SSD. The fastest we're able to read from it using the native command queuing which I was talking about before, is 528 MB/sec. So when you consider that 600 is a theoretical maximum with no other overhead, 528 MB/sec, not bad. But the Samsung SSDs properly honor the read verify, and they do real work when asked to.

So SpinRite will be able to scan any of those drives, and we've benchmarked that now at more than 800 MB/sec. We measured for this particular hardware 806 MB/sec, which means that the entire 500GB drive can have its media scanned in less than 10.5 minutes. It was actually 10.34, we calculated. So that's going to be very cool. I mean, that means even that multi-terabyte drives will be under an hour to do a full media scan with all of SpinRite's ability to really get down at the media. So we're making great progress.

We're currently tracking down an obscure but reproducible behavior that only appears to affect some HP desktops with their BIOS with a particular setting, but it does happen to be the default. So as soon as the podcast is finished this afternoon, I'll be returning to that. We've got a terrific group of very patient testers, and we're having a great time nailing down the operation of this code, which will be incorporated into the next version of SpinRite. So all making great progress and having a good time.

Leo: Must be fun for you to come up against these roadblocks and figure them out, and then the next one, and slowly work your way through it, which is really cool.

Steve: Yeah, yeah. We've had some neat comments from people who are just like people watching this process in the newsgroup, just amazed at, like - and that's always been my approach is we're going to make this work for every single possible instance. And then when it launches, it works.

Leo: Yeah. It's a different kind of coding because, although I imagine all Windows coding is somewhat like this, you have such a heterogeneous environment you're working in, you can't just come up with an abstract answer and be done. You have to test it against all these weird combinations of hardware, software, BIOSes.

Steve: It is true. Although SQRL also, the client has been out now for quite a while. No bugs.

Leo: Yeah.

Steve: So it can be done.

Leo: It can be done. All right.

Steve: Okay. So first of all, Leo, shortly after the surprise publication, which was our topic last week...

Leo: That's right, yeah.

Steve: ...of Pierre Kim's and Alexandre Torres's findings of those seven severe and apparently deliberate egregious vulnerabilities, including hard-coded username and password backdoors for the device's telnet server, C-Data, the manufacturer whose name was on those devices, posted a statement on their website. And I have to say I was very impressed with their response. It was titled "Statement on Pierre Kim Revealing Security Vulnerabilities in C-Data OLT Products."

They said: "C-Data noticed that Pierre Kim released security vulnerabilities in C-Data OLT on the GitHub website. C-Data immediately started investigation and analysis. We will give report as soon as possible. C-Data adheres to protecting the ultimate interests of users with best efforts and provide customer with safety products. Here, we express our appreciation for Pierre Kim's concern on C-Data products." And considering the truth, this is probably the most gracious statement I've ever seen.

Okay. So then at the bottom of that there's a link to the technical statement on this. So they said: "Statement on Pierre Kim Revealing Security Vulnerabilities in C-data OLT Products." Get this: "We have noticed an article named 'Multiple vulnerabilities found in C-Data OLTs' published in GitHub. C-Data admires the work of two professionals in technological circles, Pierre Kim and Alexandre Torres, and thanks for their identifying security breach problems through detailed testing, as well as for their active work in reducing the risks of users using network products. C-Data adheres to the philosophy of serving customers and always puts customers' interests in the first place, as well as pays special attention to the product safety problems. In this way, C-Data can provide customers with products with safety guarantee.

"In the meantime, we have paid attention to some press releases published by the media, and have interpreted technical articles by Pierre Kim and Alexandre Torres. In order not to let the majority of customers misunderstand the safety design of our equipment, C-Data analyzes and clarifies the mentioned technical issues with a sincere and frank manner."

First topic: "Excluding counterfeit products. The login account mentioned in this article" - and then they cite one of those four that we talked about, panger123 for the username, suma123 for the password. "We have investigated the account and password. In addition, we have confirmed that the account and password are not from the C-Data OLT products, but are those used by other companies and people when they copy the C-Data OLT hardware."

Leo: Oh, my god.

Steve: Yeah.

Leo: These are counterfeits.

Steve: Yes.

Leo: Wow.

Steve: "The CLI style [Command Language Interpreter] and most of its commands of the counterfeited OLT are all copied from the C-Data OLT. C-Data OLT equipment is now widely used around the world, and counterfeiters copy C-Data OLT for illegal profits. According to the following screenshot, we can completely compare and analyze that the account of panger123/suma123 comes from an illegally copied OLT."

Leo: Wow.

Steve: And then they show both. And I do have that second screenshot later in the show notes because it shows telnet being used to log in with panger123, password suma123. And then it says, well, it's misspelled, but it tried to say "entry superuser successfully" logged into the counterfeit system. So unfortunately, these security guys didn't know about the counterfeiting problem that C-Data is all too aware of, and so unfortunately incorrectly accused C-Data of this behavior. But there's more.

Leo: So they were testing counterfeit devices.

Steve: Yes.

Leo: Not the real deal. Wow.

Steve: Yes.

Leo: That's an interesting conundrum for security researchers.

Steve: It really is.

Leo: Wow.

Steve: Yep. So the second point, "Introduction to several factory settings accounts." They said: "The following two telnet login accounts and passwords mentioned in this article are actually" - that is, by Pierre and Alexandre - "are actually used on the C-Data's first generation OLT, OLT starting with FD11XX." And then they cite debug/debug124, root/root126. "This account and password are mainly used by C-Data to assist customers in debugging problems and writing production parameters." For example, they said: "(OLT MAC address information and Serial Number information.)"

Get this: "These accounts" - they said "this account," but they meant these accounts - "must be successfully logged in to the console port by the local serial line plugged into the OLT. Then can entering the OLT bcm-shell mode to modify and view key information of the OLT. Using this account under OLT telnet mode, we can only enter the CLI of the device. We cannot enter OLT bcm-shell to modify any information of OLT."

In other words, those telnet usernames and password combinations which the security researchers found in the firmware can only be used when connecting to the device's hardwired local physical serial port, and even then only allow for viewing of the device's fixed information such as its network MAC addresses and its serial number. You cannot get into the command shell and see anything or make any changes using those. So again, this is standard default login, and that can be changed.

They said: "If attacks want to enter the bcm-shell mode of OLT to obtain device privacy information or implant malicious programs into OLT, they must log into OLT by directly connecting the serial port line to the computer locally. In this way, by no means can remote attackers use these two accounts to attack. Therefore, there is no such situation as backdoor access with telnet."

And then I have a screenshot here, and I have it in the show notes, showing their authentic OLT device running the current firmware. And an attempt was made after logging in with root and then root123, they tried to go into the shell. And the response was only the console support is the response, meaning you can't get there from any network attached to the device, only by plugging a serial connection into the port. And that's also standard.

I mean, I'm a Cisco user. I've got Cisco equipment. And there are many things you need to do. You've got to click into their little RJ45 connector with this funky blue Cisco cable

in order to get it to an RS232, then plug it into a serial port. So these guys have done everything right. And they also talk about the third account, guest/ with no password, which, you know, is like, really? On that they say the account and password are the account of factory default configuration, which can only check some basic information of OLT, and without having the authority to configure any OLT. The user can delete or modify the account as needed when using it.

And finally, they said: "More secure cryptographic mechanism. For other models" - because remember the other critique was HTTP and no use of crypto. "For other models of C-Data OLTs named FD15XX, 16XX, 12XX, and 8000, the problem of backdoor access with telnet does not exist because these OLTs adopt a more secure cryptographic mechanism. The device is configured with several general accounts by factory default, including root/admin, admin/admin, and guest/guest, which can be used by customers to initially configure OLT. Customers need to create, delete, and modify the login account and password of the device according to their own security policies when using the device. We do not recommend using the factory default username and password in the operation network.

"The device retains a debugging account for assisting customers in debugging and solving problems, and this account can also be used by customer to find the forgotten password when they forget the login password of OLT. However, the account no longer uses the general password, and the password is calculated and generated according to the unique identification information of the customer's OLT. Only when the customer provides the information of unique identification code in conjunction with a special password generation tool can the password be generated. The password of each OLT is different, which will better ensure the safety of the device."

And C-Data's quite polite under the circumstances correction of the record continues like that for some time. So as a result of the fact that Pierre Kim and Alexandre Torres misinterpreted what they saw and were using a counterfeit device at some point, and because they misinterpreted it, they chose not to first confront C-Data with their findings...

Leo: And therein is the problem.

Steve: Yes. Their vulnerability report was full of glaring inaccuracies. The good news is that their mis-disclosure didn't actually put all of those C-Data customers' networks at risk because they never were at risk. C-Data understood that any default access credentials need to be constrained to the device's local serial configuration port. And that's the way the authentic device works. But purchasers of counterfeit C-Data equipment were not so fortunate.

The screenshot below, which is the one I referred to before in my show notes, depicts a successful login to the counterfeit C-Data device over the network - not locally, but using the panger123/suma123 username/password credentials to which the counterfeit device declares super (S-U-P-P-E-R) uer (U-E-R) successfully, presumably attempting to say "superuser," meaning full god mode access to this thing, full remote admin access granted using the secret credentials buried in the firmware of the counterfeit high-end C-Data equipment.

So where does one purchase counterfeit equipment? Certainly not from C-Data, nor from any reputable reseller. Maybe this is the stuff that's found on eBay for a bargain. I don't know. So it brings us to a really interesting issue that we haven't touched on in nearly the 15 years of this podcast, which is counterfeit networking equipment, which turns out to be a real problem and a real thing.

F-Secure Labs just published a beautiful piece of their recent research - it was dated July 2020 - titled "The Fake Cisco: Hunting for Backdoors in Counterfeit Cisco Devices." I have a link to the Cisco PDF and to F-Secure's announcement of it. And the PDF, this fake Cisco PDF is so cool that I wanted to make it easy for anyone to take a look at who was curious. So I used this podcast episode's number as the grc.sc shortcut. So grc.sc/776 will redirect you to this very cool PDF of Cisco's complete analysis of two different counterfeit Cisco routers.

And to kind of give us a sense for the reality of the world of counterfeiting, they said in their introduction: "Producing counterfeit products is, and always has been, a great business if you don't mind being on the wrong side of the law. There's no need to invest" in all that costly and, you know, well, they didn't say, I shouldn't editorialize. Reading just what they wrote: "There's no need to invest in a costly R&D process, and no need to select the best performing and looking materials. The only criterion is the cost of manufacture. This is why we see many imitations of expensive products on the market and are likely to continue to see them being made and sold at a fraction of the original's price."

You know, as I mentioned at the top of the show, like Rolex watches are famous, I mean, that's just like a clich of counterfeit now. And women's handbags, and I guess shoes and all kinds of stuff are counterfeited. And I've heard that there are DVD retailers in Hong Kong where, I mean, any movie you could ever imagine wanting for pennies on the dollar, and it looks like an absolute, you know, you can't tell by looking at it that it's not the real deal.

So anyway, continuing, F-Secure said: "Network hardware designed, manufactured, and sold under the Cisco brand is a perfect example of this. Having an excellent reputation because of their great engineering, these products sell at a premium price point. Naturally, this encourages some to try and produce counterfeits as it's a way of making easy money. Stories of such exploits abound in the media: a gang reportedly exporting $10 million U.S. worth of gear to the U.S., the FBI seizing shipments of fake hardware, and court rulings being issued to stop the manufacturers.

"What does Cisco do to combat fraud? Actually, a lot. Cisco has a dedicated Brand Protection organization whose purpose is to defend against counterfeit and gray market activities. They partner with customs teams and regional governments all over the world with success. In April 2019 they seized $626,880 worth of counterfeit Cisco products in one day. However, despite successful operations, Cisco has not been able to stop fraud fully. If there's an opportunity to make a fast buck, there'll always be someone willing to take the risk.

"In fall of 2019, an IT company found some network switches failing after a software upgrade. The company would find out later that they had inadvertently procured suspected counterfeit Cisco equipment. Counterfeit devices quite often work smoothly for a long time, which makes it hard to detect them. In this particular case, the hardware failure initiated a wider investigation to which the F-Secure Hardware Security team was called and asked to analyze the suspected counterfeit Cisco Catalyst 2960-X series switches. This initiated a research project with the following goals: Verify no extra functionality such as backdoor access was introduced; understand how and why counterfeit devices bypass the platform's authentication security controls.

"Naturally, it's not easy to tell genuine and counterfeit devices apart. To verify whether any kind of backdoor functionality existed was also not easy, as it required a considerable amount of technical investigative work. Ultimately, we concluded with a reasonable level of confidence that no backdoors had been introduced. Furthermore, we identified the full exploit chain that allowed one of the forged products to function, a previously

undocumented vulnerability in a security component" - actually it had a race condition - "which allowed the device's Secure Boot restrictions to be bypassed."

So the full report is 39 pages, and I found it utterly fascinating. As I mentioned, grc.sc/776, this episode number, will take you there. And Leo, on page 16 of the show notes I took from two separate pages of the PDF, I put the two images of the real and the counterfeit Cisco logic board, at least part of it, side by side. And, I mean, it's just a copy. You can see some slight changes in some component choices, looks like some smaller components were placed on the switch interface chips. They're larger on the Cisco on the left, smaller on the copy on the right. But, I mean, Cisco uses PowerPC as their main driver. Its heat sink is a little bit larger. But again, it looks like the circuit board was...

Leo: You couldn't, on surface examination, you couldn't - there's no way to know. I mean, it looks like the same circuit board, practically.

Steve: Yeah, it's got Cisco's, not surprisingly, Cisco's name and number. I guess there's a...

Leo: The holographic thing.

Steve: Above the processor, yes, the holographic thing is missing.

Leo: Yeah. That's what you want to see.

Steve: But you wouldn't know from the other board that it was supposed to have one.

Leo: Right, there's circuitry there, yeah.

Steve: So again - yeah. You would open it up and, like, there's nothing to see.

Leo: Always look for that holographic sticker. That's why they do that. That's apparently hard to counterfeit, I guess.

Steve: Ah, interesting.

Leo: Maybe, yeah.

Steve: I wonder why it would be hard. But you're right.

Leo: Yeah, I wonder why, too. I know they do it in money, too, so there must be something about it, yeah.

Steve: Yeah. So as it happens, though, F-Secure closely examined two devices that were both Cisco counterfeits. They were both running authentic Cisco firmware and software. So here the bad guys' goal was not to introduce a backdoor. They just wanted to sell a knockoff at a somewhat reduced price, at probably a much, I mean, it looks like it would cost them, depending upon their production facility, I don't know what quality control Cisco adds. Maybe they use cheaper fans and a cheaper power supply.

But, you know, a PC board is pretty much a PC board. And if it works, it works. And even as F-Secure said, hey, you know, these counterfeits typically work just as well. But it was an update to the firmware that caused them to be caught out because the workaround for the firmware authentication, the Secure Boot technology essentially, broke when the product was updated. So they were both running authentic Cisco firmware and software at purchase.

The first counterfeit contained add-on circuitry which exploited a race condition in the boot ROM code to bypass its software verification. It did this by intercepting EEPROM control signals, replacing certain bytes in the image being loaded to modify the software's behavior on the fly. It appears the processor's printed circuit board in this unit was an exact copy of Cisco's without modification. So they sort of grafted it. And actually it's on the underside of the PCB so you don't see it unless you take the whole thing apart and look at the bottom side, where you would say, hey, what's that little turtle, black turtle with the wires all over the place? There is a picture of it, of the underside of the circuit board in their PDF.

When the first counterfeit received what amounted to a post-manufacturing add-on circuitry, the printed circuit - whereas. Yeah, there you're showing it now is the graph that was made to do an EEPROM intercept on the fly, which is very clever, yeah. Okay. So the first counterfeit received what amounted to a post-manufacturing add-on circuitry, and no change at all to the Cisco circuit board, which was interesting. I mean, it is a clone copy of the PCB. But the printed circuit board design of the second counterfeit was changed to incorporate that hack made to the first counterfeit, which replaced the EEPROM completely with an unknown integrated circuit.

This signified to F-Secure that a considerable resource investment had been made in design, manufacture, and testing of a forged product like this, compared to the lower cost ad hoc approach used by the first counterfeit. The board layout and silkscreen labeling similarities also suggested that the people behind the second forgery may have either had access to Cisco's proprietary engineering documentation, such as printed circuit board design files, in order to be able to modify them, or they invested quite heavily in the very complex process of replicating the original board design files in order to modify them, in order to modify the actual circuit board wiring, in order to create this. And you know, it sort of brings up a point, too. Cisco probably manufactures this stuff in the Far East, I would imagine.

Leo: Yeah. It may well be made in the same factory.

Steve: Uh-huh.

Leo: Yeah, that's my guess. Yeah.

Steve: They turn the production line back on at night.

Leo: Mm-hmm. Mm-hmm.

Steve: And run off a few more copies. So in the case of the C-Data counterfeit, a seriously dangerous, remotely accessible backdoor was definitely installed into the counterfeit devices. In the case of the extremely elaborate Cisco counterfeits, all the ingenuity was expended in creating a virtually indistinguishable clone of the original and then engineering around Cisco's detection that its prized network operating system was running in counterfeit and unauthorized hardware. So an interesting tale of two counterfeits.

Leo: Amazing, yeah. Really amazing, yeah. Wow, what a great story. And what a great show. Thank you, Steve. Security Now! airs every Tuesday, if you want to watch the live version of the show. Start time's a little tricky. It's the third show of the day. Things get pushed a little bit sometimes. But usually we're trying to hit 1:30 Pacific, 4:30 Eastern, 20:30 UTC. If you tune in and MacBreak Weekly's still on, well, just stay tuned, and Security Now! will come along eventually. Just watch it at the stream which is twit.tv/live. There's audio and video there.

After the fact, the shows are available on Steve's site, as well as our site. Steve's site is GRC.com. That's where you'll find out about SpinRite, the world's best hard drive maintenance and recovery utility. And you can get your copy of 6.0 ready for 6.1, an automatic upgrade if you do that now. Plus you can participate in the beta tests and all the things he's trying out.

While you're at the site, 16Kb versions of audio for the show are there, as well as 64Kb audio, as well as transcripts. And a lot of people like those transcripts, and I use them all the time for search because, if you're searching, you can almost always find what you're looking for in the transcripts and jump to that part of the audio. So that's GRC.com. Steve's also on Twitter. He's @SGgrc. And that's a good place to go if you want to message him. He takes direct messages and engages at Twitter: @SGgrc. Or you can go to GRC.com/feedback and leave something on the feedback form.

Our site is TWiT, of course, TWiT.tv/sn for this show. You'll find copies of audio and video there. We do video, as well. You'll also find it on YouTube, the videos on YouTube. There's a Security Now! channel. Best thing to do, get a podcast program and subscribe so you get the episode the minute it's available. And you can begin your collection. Collect all, what is it, 776. Get the complete set of Security Now! episodes. And Steve, 777 next week.

Steve: Ooh, cool, yes.

Leo: Yeah, it'll be fun. Good luck number if there ever was one. Thanks, Steve. We'll see you next time on Security Now!.

Steve: Thanks, buddy.

Copyright (c) 2014 by Steve Gibson and Leo Laporte. SOME RIGHTS RESERVED

This work is licensed for the good of the Internet Community under the

Creative Commons License v2.5. See the following Web page for details:
http://creativecommons.org/licenses/by-nc-sa/2.5/