Ripple20

Description: This week we look at Microsoft's interesting decision to update Windows 7 desktops with their new Edge browser, Google's wholesale removal of 106 widely downloaded malicious Chrome extensions, Microsoft's continuing drama over Win10 printing, a potentially critical remote code execution vulnerability in everyone's favorite VLC media player, an interesting move by Roskomnadzor, Netgear's residence in the doghouse, a new and startling record in DDoS attack size, a bit of errata, and the anticipated announcement of a new piece of spin-off freeware from the SpinRite project. Then we examine the ripple effects of the mass adoption of an embedded TCP/IP stack that is found to be horribly insecure many years after it has been quite widely adopted across the embedded device industry.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-772.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-772-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with lots of security tales of woe including 106 malicious Google Chrome extensions, the truth about Zoom end-to-end encryption, and then we'll talk about two massive flaws - one in more than 70 Netgear routers, the other in pretty much every IoT and embedded network device anywhere. And there's not much of a fix. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 772, recorded Tuesday, June 23rd, 2020: Ripple20.

It's time for Security Now!, the show where we cover you with privacy, we wrap you up in security, we pat you on the head, and we put you to bed. And it's all done...

Steve Gibson: And send you out trembling.

Leo: Trembling, yeah, actually that's more like it. There he is, the man of the hour, Steve Gibson of GRC.com. Hi, Steve.

Steve: Leo, great to be with you again. We have a rip-roaring show. It's titled for this Security Now! 772 "Ripple20," which - ah, boy. This speaks to why we really don't want to have a technological monoculture. But lots of news. We're going to look at Microsoft's interesting decision to once again update Windows 7, but not with what you would expect, not security, but with the Edge browser.

We're going to look at Google's wholesale removal of 106 widely downloaded malicious Chrome extensions and what brought that to their attention. Microsoft's continuing drama over Windows 10 printing not resolved yet, folks. A potentially critical remote code execution vulnerability in everyone's favorite VLC media player. An interesting move by, wait for it, Roskomnadzor, of course Russia's media watchdog.

Leo: We're going to really have to get a Russian to pronounce that for us so we know how that goes.

Steve: Yeah, maybe. Although I'm having too much fun with it as it is. We also have Netgear taking up residence in the doghouse. Oh, goodness. Also a new and startling record has been broken in DDoS attack size. We've also got a bit of errata as a consequence of you and I discussing how long we've been doing the show last week. That created quite a kerfuffle over in GRC's SpinRite newsgroup, or Security Now! newsgroup.

We're going to - oh, I also have the anticipated announcement of a new piece of spinoff freeware from the SpinRite project. And then, time remaining - actually, no, we'll have time - we're going to examine the ripple effects, thus the name Ripple20, of the mass adoption of an embedded TCP/IP stack that is found now, 23 years since its birth, to be horribly insecure and quite widely adopted across the embedded device industry.

Leo: Huh.

Steve: Yeah. And we do have sort of a unique Picture of the Week for those who are curious.

Leo: Yes. I can't wait to show everybody.

Steve: So anyway, you were talking on Sunday on TWiT about your blog, and you showed everybody a picture of yourself, I'm not sure at what...

Leo: In high school.

Steve: Oh, in high school.

Leo: In high school, yeah.

Steve: So as it happened, one of my gang, my high school gang that I've referred to a couple times, we've maintained a connection to each other, and we're in a group text. I was listening to you guys talking on MacBreak Weekly about group texting. I thought, yeah, I'm in a couple groups.

Leo: Nice.

Steve: Anyway, Jim - who was my best friend at the latter half of high school, and so when we both got into Berkeley, we requested each other as roommates, and he became my roommate - is in the process of digitizing his vast paper printed photo collection back from the Eastman Kodak days. And he ran across some that he had taken in 1973 and shot them to me. Anyway, so since this happened a couple weeks ago, I thought, oh, I have a picture. Anyway, so for what it's worth, I mean, it's not going to be of interest to most people, but...

Leo: Oh, it's of great interest. Are you kidding? This is you in college? Are you a freshman? What year are you?

Steve: I was a freshman.

Leo: As all freshmen do, you had stolen a street sign. I think that's good.

Steve: I had my, yes, my required street sign, and posters on the wall.

Leo: Posters, yeah.

Steve: And I love the out-of-focus in the foreground coffee pot.

Leo: Coffee pot, we know where that started.

Steve: And I still remember that particular coffee pot. That was a very important item in my life at the time.

Leo: Oh, that's great. I love it.

Steve: And there is, in the upper left, you can sort of barely - I know what it is because I remember it. But I was messing around with plotters and the Xerox graphics printer. And so there are some vestiges of things from SAIL, Stanford's AI Lab at the time. And, you know, just crazy stuff. And then there's that cranky woman whose picture is covering up the sign.

Leo: We've all seen that.

Steve: I don't remember whether I took it or Jim took it. But I was messing with black-and-white and Ilford film at the time.

Leo: That's funny. So it's your picture, huh?

Steve: And so I might have just, I mean, that was an actual woman in Berkeley who was really not happy to have her picture taken.

Leo: That's hysterical. That's hysterical.

Steve: By some random kid. But she was just such a grouch, I put her on the, you know, I thought, well, I'll make an enlargement and have fun with that. So anyway...

Leo: You also have the requisite "Danger" sign. Everybody should have that. I mean, this is good. This is good. I love this. This is so classic. I know this era. What was it, '71, '72, thereabouts, somewhere like that?

Steve: Yeah, it was 1973 is when I graduated from high school. So it would have been toward the end of '73 that I was a freshman at Berkeley.

Leo: Wow. Thank you, Steve. You made my day.

Steve: Anyway, just a little blast from the past.

Leo: It is awesome.

Steve: And I haven't changed a bit since then.

Leo: A bit. You know what, you were actually a little chubbier then; you know? You've leaned out a little bit.

Steve: Yeah, I had a little bit of meat on me back in those days.

Leo: Yeah. But that moustache, that's the same one.

Steve: And like the sideburns, Leo? They come right down and wrap around underneath my jaw line there.

Leo: Yeah, love them, love them. Very impressive. I had your hair, too. I had that exact hair. It's so funny.

Steve: Everybody did back then.

Leo: It's so cute.

Steve: Now none of us do.

Leo: Right.

Steve: So starting last Wednesday, the 17th, our long-dormant Windows 7 Update suddenly sprang back to life as Microsoft began rolling out an "important," is what it was labeled, update for their long-since-forgotten Win7 64-bit systems. It's 4567409, titled "Microsoft Edge Update for Windows 7."

Leo: Oh. Oh. Oh.

Steve: I know. It's like, what? "For x64-based systems." And of course that's their new Chromium-based Edge web browser. The bulletin states that this update is not being offered to enterprise devices, only to users running Windows 7 SP1 and Win8.1 Home, Professional, Ultimate, Starter, or Core editions. And they wrote: "This update is not intended to target Enterprise devices. Specifically, this update targets devices that run Windows 7 SP1 or later versions and Windows 8.1 or later versions that are either Home, Professional, Ultimate, Starter, or Core. Devices that run these editions on Active Directory or Azure Active Directory domain are also excluded from this automatic update." So in other words, all the rest of us.

They said it's only available via Windows Update and not being offered as a standalone download from the Microsoft Update Catalog, as opposed to, as we'll see in a minute here, some things that I wish were being offered by Windows Update, but aren't. And they said: "After being installed, the following changes will be made." And this, again, this is unsolicited; right? "Edge will be pinned to the taskbar and add a shortcut to the desktop. If the current version of Edge, if any, already has a shortcut, it will be replaced. The new Edge will not replace IE, and this update will not change the system's default URL handler."

So that's nice. It's not like suddenly it pops up when you click a link, if you've got Chrome or Firefox on Windows 7. But still, it's interesting; you know? It suggests that Microsoft would like to have a bit of a foothold in all of those non-enterprise machines that are still stubbornly running Windows 7. And I was curious like what that meant. The latest market share has Windows 10 solidly in - that is, market share of the desktop - Windows 10 solidly in first place at 53.74%. Windows 7 is in solid second place at 28.35. But after that, so Windows 10 at nearly 54%. Windows 7 at about 28%. Then it's a drop to third place for macOS X v10.14 at, get this, 3.84%. Then Windows 8.1 at 3.65.

So together the pair of Windows 10 and Windows 7 total 82% of all desktops. So it's still going strong. And I guess I can see why Microsoft might not be willing to concede the desktop browser to Chrome when it no doubt feels that it now, finally, has something, thanks to Chromium, of course, that's every bit as good. But anyway, it's just kind of weird that, yeah, we're not going to fix any security problems you have; but we think you should have the Edge browser on Windows 7 just because. It's like, okay.

Leo: Wow.

Steve: Yeah, isn't that weird?

Leo: Of all things to update. That's so weird.

Steve: I know.

Leo: You know, I had a call this weekend on the radio show by a guy who said that happened. And I was skeptical. I thought, no, why would Microsoft do that? It makes no sense.

Steve: Doesn't that sound completely random? It's like, what?

Leo: Yeah, well, they were.

Steve: Yeah. And I did already because of the podcast. I already had Edge on my Win7 machine that I'm sitting in front of right now. And so I didn't notice any change. I did go into it to see if it said anything about update pending or a new version or something. But I didn't catch anything. But anyway, I thought that was just sort of curious. And certainly there are, what is it, about one third, no, half as many Windows 7 now as Windows 10. So it's beginning to ebb. And you would expect that, now that security updates have once and for all finally really truly been terminated. So, yeah, it's pushed people over to Windows 10. And I'm sort of making a fragile peace with it.

I tried to install Windows 10 over the weekend on an old x86 machine, the very end of the work that I was doing on this new InitDisk utility. Somebody managed to get it to pop up a complaint about the floppy drive not being ready on one of his systems. And I was unable to make it happen. And then he explained, oh, no, you have to have a diskette controller on a motherboard with no disk drive, no floppy drive plugged into the controller. If it's plugged into the controller, no problem. But if it's a floppy controller that doesn't have a floppy drive on it - anyway, to make a long story short, I fixed that. It no longer does that. But in order to verify it, I had to install Windows 7 on an old machine that still had a floppy controller. Turns out that they're kind of rare now. I mean, it's easy to plug in a USB floppy drive, and most BIOSes know how to support that.

Leo: Those are pretty rare, too. I'm amazed you could find one.

Steve: Yeah, they are. But then I wanted to try Windows 10 because I figured, well, while I'm at it, I might as well be able to boot this old x86 machine into 7 or 10. It kept refusing to take a Windows 10 install. Finally, I cloned the x86 Windows 7 install to another drive. Then I used that installer updater thing, I can't remember the exact name for it, which runs on x86. It came up under Windows 7, said hi, would you want to update to Windows 10? And I said, yes, because nothing else, it works.

So it downloaded what it needed to and churned away for a while and rebooted a few times and finally said, this Pentium 4 doesn't support NX. Windows 10 will not run. So it's like, oh, that's interesting. So that's of course the no-execute bit, which was added later to the Pentium 4 than my chip, apparently. And so no go. So anyway, I'm happy with Windows 7 on that. And I'll be talking about InitDisk a little bit later. And I have no idea how I got off onto this.

Leo: Well, but just to be clear, you need to do this because of SpinRite. You've got to have all this legacy stuff lying around; right?

Steve: Exactly. That's precisely why. Yeah. And the technology that I'll be talking about later is a spinoff of SpinRite because I needed an absolutely bulletproof - well, I'll talk about it in a minute. Later. We've got lots of other stuff to talk about.

Leo: Yes.

Steve: Google yanked 106 out of 111 - apparently five somehow they didn't agree with - malicious Chrome extensions which were found to be collecting sensitive user data. The well-named cybersecurity firm, Awake Security, because that's the way you want your cybersecurity firms, identified the extensions as malicious in a report they published last Thursday which they titled "The Internet's New Arms Dealers: Malicious Domain Registrars." And of course you and I, how many times have we talked about the danger of DNS registrars doing the wrong thing? You know, domain name registrars issuing certs that they shouldn't.

In this case, they explained that their report dives into the results of their multi-month investigation that uncovered a massive global surveillance campaign affecting millions of users, like 33 million. The campaign involved thousands of domains and more than 100 malicious Chrome extensions with all the activity tying back to a single Internet domain registrar, GAL (G-A-L) Communication. And they have in here, they have CommuniGal (C-O-M-M-U-N-I) Ltd., also known as GalComm. They said: "This campaign and the Chrome extensions included operations such as taking screenshots of the victim device, loading malware, reading the clipboard, and actively harvesting tokens and user input. In the wake of Awake's disclosure, Google has taken down the malicious extensions" - all but five. "However, the campaign was able to avoid detection by using state-of-the-art security tools through a number of evasion schemes."

So this report highlights that, they said: "The attacker's infrastructure, including 15,160 malicious or suspicious domains and 111 malicious or fake Chrome extensions, collectively had approximately 33 million downloads." They said that connections between the campaign and a number of traditional malware families had been seen. The methods the attacker used to avoid detection by sandboxes, endpoint detection and response systems, web proxies and more they also got into; and the way Awake connected the dots leading back to a single domain registrar. They said that the extensions posed as tools to improve web searches, to convert files between different formats, to perform security scans, and more.

But the extensions contained code to bypass Google's Chrome Web Store security scans, thus to get into the store in the first place. Then, once downloaded, as had happened 33 million times, to take screenshots, to snag the global clipboard, to harvest authentication cookies, and to snag user keystrokes including passwords. They said that they believed all the extensions were created by the same entity, although the company has not yet identified who that entity is.

The primary connection between all the extensions was that they sent user data back to domains that were all registered through the GalComm domain registrar. And they said that many extensions appeared to share the same graphics and code base with only slight changes. That similarity made scanning for their sibling extensions much easier. Some of the extensions even shared version numbers and descriptions that no one had bothered to change from one to the other because they were just basically spitting these out, trying to sneak them past Google, and it had become a numbers game.

So anyway, someone had gotten lazy about covering their tracks. You know, and I have two screenshots from the Chrome Web Store of two samples. One calls itself ByteFence Secure Web Browsing. Okay. And they all did have lots of faked user feedback, you

know, five-star browsing extensions, you know, the best antivirus software I have, or I've ever had or something. You know, complete nonsense. Also another one, Secured Search Extension. It's like, you know, what? Who knows? But people were downloading it because, oh, look, I'd like to have my searches extended or secured.

So Awake said that by last month, when it reached out to Google, having finished its research, the 111 malicious extensions they had identified had been downloaded, oh, not quite 33: 32,962,951 times. And I suppose the rule here is if you build it, they will download it. And indeed that happened. Based on their internal telemetry, Awake says that some of these extensions have been found on the networks of financial services, oil and gas, media and entertainment, healthcare and pharmaceutical companies, retailers, high tech, high education, and government organizations. They are currently - well, they were - acting as backdoors into private works and as espionage tools. Though they have no direct evidence to suggest that they've ever been used that way, the capability is there in them.

So after being notified, Google has so far proactively deactivated 106 out of the 111 identified Chrome extensions. And they've been, you know, so they're obviously removed from the Web Store. But they've been actually turned off in every user's browser. The extensions are still installed, but they've been disabled and marked as malware. So if you were interested, you go to chrome://extensions, which is the way your Chrome browser will show you what's installed. And if it's been turned off, there's like a red "malware" that is saying, you know, we turned this off to protect you. Also, if you see that, you should take a lesson from the fact that basically this is a numbers game, and you don't want to get bitten by this.

And it's probably a good moment for us to just mention that, you know, add-ons of all types have become the new front in the war on consumer trust. We know that Apple and Google invest heavily in the safety and security of third-party apps and add-ons that they curate in their various app stores, but that nevertheless stuff is going to slip past. Many people choose an Android device because, as we know, they are typically less expensive, and often because of the greater user freedom that comes with a more open platform. And perhaps like your daughter Abby, Leo, they just don't like Apple for whatever reason.

Leo: Right.

Steve: So they're going to give their money to Samsung or Google or another hopefully major Android brand because we know how important it is that these things get updates. But the same openness which is given to the user is also available to the apps. And the user is trusting those to be safe. So the bottom line is, with these deals where you've got add-on apps and extensions, is that it is a pure numbers game. The chance that any one particular app might be malicious is probably small because these companies are doing a good job. But every additional app or add-on which is installed increases the probability that one of the growing collection is probably malicious.

So it's prudent not to let the crap accumulate in your browser with extensions or in your smartphone with add-on apps. It's useful to occasionally page through what you've downloaded and remove the things that you no longer use. Even though, Leo, I did hear you last podcast indicate that...

Leo: I like to have that stuff.

Steve: ...you've got things you haven't touched for years, and you do not want them to be silently removed from your phone.

Leo: No. But you can turn that off, so I do.

Steve: That's true.

Leo: Yeah.

Steve: Anyway, just download stuff with caution. I have a few things. I don't download add-ons in my browser because it is the case that every time you do something like that you're taking a slight risk, and they do add up.

Okay. Zoom's encryption story, take three. And I'm being kind only saying three.

Leo: Uh-oh, yeah.

Steve: Last Wednesday in an apparent effort to deal with his company's recent spate of confused mixed messages regarding Zoom's actual plans for enforcing truly secure end-to-end encrypted teleconferences, Eric Yuan attempted to clean things up in a blog. Eric wrote exactly the following: "Since releasing the draft design of Zoom's end-to-end encryption on May 22nd, we've engaged with civil liberties organizations, our CISO council, child safety advocates, encryption experts, government representatives, our own users, and others to gather their feedback on this feature. We've also explored new technologies to enable us to offer end-to-end encryption to all tiers of users. Today, Zoom released an updated end-to-end encryption design on GitHub." And I looked, and it was just like commas and dots and things were changed, no biggies.

He said: "We are also pleased to share that we have identified a path forward that balances the legitimate right of all users to privacy and the safety of users on our platform. This will enable us to offer end-to-end encryption as an advanced add-on feature for all users around the globe, free and paid, while maintaining the ability to prevent and fight abuse on our platform."

Leo: Yeah, there's your clause.

Steve: Uh-huh. "To make this possible, Free/Basic users seeking access to end-to-end encryption will participate in a one-time process that will prompt the user for additional information, such as verifying a phone number via a text message. Many leading companies perform similar steps on account creation to reduce mass creation of abusive accounts. We're confident that by implementing risk-based authentication, in combination with our current mix of tools including our Report a User function we can continue to fight and prevent abuse."

Then under "additional information" he said just four bullet points: "We plan to begin early beta of the end-to-end encryption feature in July. All Zoom users will continue to use AES-256 GCM transport encryption as the default encryption, one of the strongest encryption standards in use today." And note that's the way they're differentiating. They're saying "transport encryption" as opposed to full...

Leo: End-to-end.

Steve: ...end-to-end encryption.

Leo: Right.

Steve: Then they say: "E2EE will be an optional feature as it limits some meeting functionality, such as the ability to include traditional dial-up phone lines or VoIP hardware conference room systems. Hosts will toggle E2EE on and off on a per-meeting basis." And actually I believe they meant only on, because they have previously said it will absolutely not be turn-offable once it's been turned on. Then finally the fourth bullet point: "Account administrators can enable and disable end-to-end encryption at the account and group level." And he finishes: "We are grateful to those who have provided their input on our E2EE design, both technical and philosophical. We encourage everyone to continue to share their views throughout this complex, ongoing process."

Leo: It feels to me like, honestly, they're trying to distract everybody from the real issue, which is is it really end-to-end encryption?

Steve: Yes.

Leo: And I'm still unclear on that. What is your opinion on that?

Steve: So my opinion is they have royally screwed the pooch, and no one is ever going to trust them again. So they're trying to say - and Alex has sort of talked about this. I'd seen references to it, the idea being that, if they can - their concern is that there's rampant abuse of encrypted teleconferences because of the abuse of anonymity. And that, if they can trade enhanced security for some reduction in anonymity, that that will put off the abusers who will not want to give a phone number, which is then tied to their conferencing.

Leo: Yeah, no, all that makes sense. But it doesn't address the end-to-end encryption part of it. It's just talking about authentication. I keep feeling like they're trying to say "Pay no attention that. Here's what we're going to do for authentication." But when they talk about this turning on this better-than-transport encryption, at the same time as they talk about that, don't they still say, but we'll be able to help law enforcement, or we'll be able to give, I mean, don't they imply?

Steve: Well, you notice there was none of that language in his blog posting.

Leo: Right.

Steve: What we know happened is there was a huge backlash from the industry at large at the idea that encryption was not going to be available for the free tier. And then there was all this back-and-forth doubletalk about it. And, you know, my feeling is, after their

out-of-the-gate stumbles, they had one chance at redemption. The very first initial problems they jumped on. But as we know, through a bizarre lack of messaging coherence, they just confused everybody. And I think...

Leo: Well, I think that's intentional is what I think. Go ahead. What do you think?

Steve: Maybe. I think they lost any hope forever of convincing those who most have a need for easy-to-use, really encrypted, warrant-proof conferencing, that Zoom is the place to get it. It just - no one is going to trust them or believe them again. And as you said, even now, there's still some confusion.

Leo: It's not clear. Is it warrant-proof or isn't it? By the way, that's a good test. And they've implied that they will help, but only in the case of child sex abuse material. But that's not the issue. And I feel like there's a lot of hand waving. I don't even think the free versus paid is the issue if none of it is end to end. And that's the real issue.

Steve: Correct. And their original issue with paid is that paying means there's a money trail, and that creates a tighter identity confirmation.

Leo: Yeah, but that's not the question I had. Maybe some people have it.

Steve: No, no, I completely agree. And but you know, Leo, what's interesting, because I thought about this, no one has any similar concern about Apple. Apple has played the encryption issue exactly right, from day one. Everyone knows that Apple's products are as absolutely and truly secure as Apple is capable of making them. Any mistakes that they or others find are immediately fixed. And Apple's very public fights with law enforcement have only proven to enhance their well-deserved reputation.

Leo: Does FaceTime use end-to-end? They don't have any - that's warrant-proof encryption in FaceTime, at least person-to-person; right?

Steve: Yes.

Leo: Apple doesn't have the keys to that.

Steve: Well, no. And I still argue that, if you're not managing the keys, somebody else is.

Leo: That's my point exactly, yeah.

Steve: And if somebody else is, that gives them the opportunity, whether they take it or not.

Leo: Right.

Steve: So nothing we've seen has shown how they could engineer themselves out of that position. And we've seen things that, like, well, iCloud isn't encrypted. And if the terrorists hadn't, or if the FBI hadn't shut down and restarted the phone, blah blah blah blah blah. I mean...

Leo: Yeah, we know Apple retains the keys to iCloud. That's known.

Steve: Yeah.

Leo: It is the case, though, that an iPhone has a Secure Enclave, generates its own keys. And so a point-to-point call in FaceTime...

Steve: Yes.

Leo: ...should be fully encrypted. There's no Apple in between. I don't know how group calling - this is the challenge Zoom faces, and everybody else, is group calling; right? Because you have to route video, unencrypted video at the server.

Steve: Right. Well, and I've listened to you talk about this across TWiT. And I agree with you that it's unclear...

Leo: It's just unclear.

Steve: ...how much Zoom-style teleconferencing really needs ber-secure end-to-end encryption anyway.

Leo: Right.

Steve: As we know, their post-COVID usage explosion is almost entirely replacing meetings such as yoga classes.

Leo: Yeah, who cares?

Steve: And traditional classrooms. They were publicly accessible and never particularly secure in the first place.

Leo: But there are dissidents. There are psychotherapists and psychiatrists. There are medical doctors. You know, HIPAA has been suspended for the nonce. So it's not even a HIPAA issue. But still there's some reasons people might want to have really private conversations on Zoom, not all of them nefarious. If you run a Jitsi server, you have transport encryption, decrypted at the server. But if you run the server,

you still retain the keys. So it is possible to do something like that, if you run your own server. I just don't - I think, it's my guess, and I wish they'd be clear about this, that they do retain the keys to it. Right? Zoom does. We've seen nothing to say they don't.

Steve: Well, the document says that their technology does negotiate end-to-end, and that they have no visibility into the keys.

Leo: Oh, okay. Okay.

Steve: So if that document is adopted, and that's sort of the question, too, is that it's not open source...

Leo: That's just - that's just - right.

Steve: It's just a design goal.

Leo: Right.

Steve: Yeah.

Leo: Well, anyway, all right. It's good.

Steve: So anyway, I just wanted to touch on this briefly. Last week we covered some of the many problems people were experiencing after installing Windows 10 June updates. Of course among them was the "printer not turned on before starting Windows," and the more significant "printing no longer works at all" after the updates. As we were podcasting, Microsoft was acknowledging some printing problems and issuing an out-of-cycle emergency update to fix one of several problems. But apparently, and incredibly, neither of the problems we talked about. There is another one which they did fix, a crash in the print spooler - that also prevents printing - in the Windows Message Center.

Microsoft says: "An out-of-band optional update is now available on Microsoft Update Catalog to address a known issue in which certain printers may be unable to print after installing updates released on June 9th, 2020." The announcement noted that the issue could cause the print spooler to "generate an error or close unexpectedly" - and of course it's never supposed to close, so any closure would be unexpected - "when attempting to print, and no output will come from the affected printer."

They also said: "And you might also encounter issues with the apps you are attempting to print from, such as receiving an error, or the app might close unexpectedly." And the issue "might also affect software-based printers, such as when printing to PDF." So pretty much everything.

And you know, as I was reading that, and this is optional update available in the catalog, it's unclear to me what users who are not proactive about managing Windows are expected to do because Microsoft recommends users to install one of the three consecutively numbered updates - 4567512, 513, or 514 - which are flagged as "optional

cumulative updates for Windows 10." And that spans versions 1909, 03, 1809, and 1803. But only if their devices are affected by that issue. And they noted that these optional Windows 10 updates are not available from Windows Update and will not install automatically.

And then, since they broke printing across all versions of Windows, two days later, last Thursday, they dropped another collection of similar optional cumulative updates to address that same printing issue on the versions that that first drop hadn't covered. I'm not supposed to say 2004 because that's confusing. 2004, 1709, 1703, 1607, and 1507. So all the way back. And Windows 8.1, Windows Server 2012, and Server 2016. So they just really messed things up. But still nothing about the "be sure to turn the printer on first" problem.

And what seems to me is that these things being optional, where the affected user must go to get the fix for the breakage is puzzling because it sure seems to me that, if Microsoft is going to automatically break somebody's Windows 10 machine, they really should also automatically fix what they've broken, especially when they've given now users no choice about whether to accept and install the forcible breakage in the first place.

I don't know. It's just a little confusing at the moment. I did see some reference to them going to be changing their Insider program and maybe rearranging things. I hope that they're able to get a handle on these things that they've been messing up recently. And maybe these are optional until next month. They didn't say that if you just waited until July, which is approaching, then they'll get all these fixed in the next cumulative rollup. I hope so because otherwise less proactive users are just going to stay broken.

I mentioned the VLC media player, or VideoLAN. I've got it installed on my machines. It's a great multipurpose media player. Version 3.0.11 fixes what they described as a severe remote code execution flaw. So worth doing. It's available now, 3.0.11, for the desktop versions - Windows, Mac, and Linux. And in addition to some random bug fixes and improvements, the main reason to update, if you're a user of VLC, is that it fixes a severe security vulnerability. And this of course follows the theme of interpreters are hard to make perfect, and unfortunately perfection is required.

It's tracked as a CVE-2020-13428, a buffer overflow in VLC's H26X packetizer. And they said, if exploited, it would allow attackers to execute code under the same security level as the user. So the good news is that it's running as a client of the OS in the user's account. It has no reach down into the kernel. The bad news is you still don't really don't want to be running someone else's deliberately malicious code in your computer.

And according to their security bulletin, this vulnerability can be exploited by creating a specially crafted file, like a malicious video which could be posted somewhere, and having the user open it in VLC. That would be the exploitation of a remote code execution. VLC would be used to read a file that had been crafted to take over your machine when that happens. So the VideoLAN guy said that the vulnerability would most likely only crash the player. They warn that it could be used by an attacker to execute code under the security level of the user remotely. But as we know, crashes are where exploits are born. So the fact that it's a crash today doesn't mean that's all it will stay.

So anyway, I would recommend anybody using VLC, just launch it. I did that last night when I saw this. I'd been running 3.0.8. And immediately upon starting it I got the news that 3.0.11 was available, and did I want to install it. So it's very quick to do, and just a heads-up that it's probably worth doing.

Also a puzzling update from Roskomnadzor, which is my American pronunciation of a probably very different-sounding...

Leo: Make it sound like Boris Badenov. You'll be okay. Roskomnadzor.

Steve: Exactly. As we've spoken of them often and had fun with their name, the bureau serving as Russia's media watchdog, okay. Here's what's weird. They said that Telegram has agreed to help Russian law enforcement fight against extremists and terrorist content shared on Telegram, on the Telegram platform. I have a link to a Russian - a gov.ru page that I just figured that my fonts all broke when I clicked it because - no.

Leo: Cyrillic.

Steve: So no help there. But I did find some coverage of this in various places. Apparently the Russian government has lifted what was largely an ineffective two-year ban on Telegram's instant messaging service, which we know is encrypted. And, boy, all you have to do is just have it show up in Russian and it's encrypted. In a message posted on its website, Roskomnadzor said it lifted the ban after Russian prosecutors reached an agreement with Telegram's founder, Pavel Durov. Russian officials said that Durov "expressed readiness to counter terrorism and extremism."

Okay, now, that doesn't actually say that Pavel agreed to drop his drawers for Russia. So we don't really know what has happened. The details about this collaboration, such as it might be, between Telegram and Russian officials has not been made public anywhere that I could find. Now, to remind our users, Russia officially banned Telegram a little over two years ago, on April 13th of 2018. That ban followed Telegram's refusal to cooperate with Russia's FSB, their federal security bureau, which is their primary intelligence service.

At the time, FSB investigators tried to obtain the encryption keys from Telegram to decrypt conversations between two suspects that were under investigation in the 2017 St. Petersburg Metro bombing. I'm sure that Pavel said, "We don't have the keys. Our system is designed end-to-end secure. Sorry, we can't help you." That wasn't the answer they were looking for. So being viewed as refusing to cooperate, the FSB filed a lawsuit which, surprise, it eventually won in the Russian Supreme Court early in 2018. Russian officials initially fined Telegram, but then the Russian courts also ordered Roskomnadzor to ban the app inside Russia. Basically, you know, we're going to shut you down, you bad Telegram people.

However, Roskomnadzor had a difficult time enforcing the ban over the past two years. Telegram constantly changed its servers' IP addresses and also employed a technique known as "domain fronting" to bypass the ban. Domain fronting is basically using different domains as a front for the actual Telegram service. And that allowed Russian users to continue using its service.

And recall that in one famous, or maybe now infamous, botched effort to ban the service in Russia, Roskomnadzor applied a wide area blanket block which affected more than 19 million Amazon and Google Cloud IP addresses, blocking out countless legitimate services inside Russia such as all of Google's services, online games, banking sites, cryptocurrency exchanges, and mobile apps. They also, in a continuing effort, banned 50 VPN and proxy services that Russians were using to access Telegram. Throughout all of this, Telegram remained extremely popular in Russia; and, despite the ban, was often used by Russian politicians themselves because, after all, secure; right? Encrypted; right? So the officials were trusting Telegram to keep their conversations safe from FSB surveillance. So again, sort of a political win for Telegram for saying no to law enforcement. Just as Apple has benefited from their own stance.

So a few days prior to Roskomnadzor lifting the ban, as they did last week, the Russian news site Znak reported that Russian members of Parliament had introduced a new bill to have the app unbanned, though it's unclear whether the bill played any role in Roskomnadzor's lifting of the ban. However, I did some more digging, and I discovered that in April of 2020 the government of Russia started using Telegram themselves to spread information related to the COVID-19 outbreak.

Leo: Ah. Well, that's what people use.

Steve: Yes. So perhaps that factored into this change. They thought, well, okay, we don't like it, but it's secure. People are using it. We have not been able to block it. Now this is getting kind of embarrassing. So let's lift the ban and make it officially legal again. I also found that Roskomnadzor had not indicated how the two organizations had been able to overcome the issue of Telegram's end-to-end encryption, if indeed they actually had. And I think they hadn't. It did not say whether it now had access to messages, or whether changes had been made to the platform. And neither Telegram nor Pavel Durov, who both regularly use Telegram to communicate with their own users, have yet commented publicly on the lifting of the ban.

So my own reading between the lines says that the Russian government, as I said, secretly regretted their opposition to Telegram, especially considering that it's the instant messaging platform of choice for many of them themselves. And perhaps the need to communicate through COVID-19 demonstrated the platform's versatility and utility; and they thought, well, since we can't actually stop it anyway, let's just legitimize it. So, cool.

Okay. This is one of the two biggies of the week. Netgear. There have previously been various other reports of random routers, even bunches of routers, having exploitable problems. But those reports haven't risen to the level required to raise an alarm for me to share with our listeners because their problems have invariably been LAN-side issues, rather than WAN-side problems. And while LAN-side issues are not nothing, they do not directly expose the router to external attack. They're only vulnerable if an attacker has already gotten themselves inside, behind the router, on the LAN, and onto the network. And of course once any attacker has accomplished that, it's pretty much the case that all bets are off.

However, when a router is weak enough on the inside, there is one troubling case where an equally weak attacker, meaning attacker that doesn't have much attack strength, might be strong enough. And that's when a user browses to a malicious website. When that happens, the attacker's code is running in the visitor's browser, which exists on the LAN. We were recently talking about how JavaScript code was able to launch its own HTTP queries using the so-called "AJAX" primitives. AJAX is the abbreviation for Asynchronous JavaScript and XML. Which, again, it allows a web page to be active, to alter its contents to establish a communication back to the originating web server and so forth.

So there are 79 Netgear models - 79 - so vulnerable, and we'll see in detail why in a moment, that just surfing to a malicious website could allow the code the browser downloads to blindly connect to the network's local Netgear router and cause it to open a telnet session, with port and command prompt as root.

Okay. So let's step back a bit. I should note that I created a link for our listeners. And Leo, you may be interested to bring it up. It's grc.sc/netgear, just so anybody with a Netgear device, you're going to want to be seeing whether your one of 79 Netgear router models may be vulnerable and may have firmware available: grc.sc/netgear. SC, of course, for shortcut.

Okay. So here's what's going on. An unpatched zero-day vulnerability exists in 79 Netgear router models. It allows an attacker to take full control over any of those 79 Netgear model devices from within the LAN, even from code running inside a user's web browser. The vulnerability was independently discovered by two researchers. Naturally it exists in the HTTPD, the HTTPD daemon used to manage the router. In other words, that's the router's web-based router management server, web server, which you typically connect to with your browser. Unfortunately, your browser can connect to it when driven by script in the same way.

One of the two researchers released a detailed explanation of the vulnerability, a proof-of-concept exploit, and scripts to find vulnerable routers. So the hackers of the world are already clued in. All of this is public. The proof of concept and the scripts are up on GitHub, and their detailed explanation is here in the show notes. So it is a fully detailed technical blog, and the posting is quite detailed. So I'm going to excerpt from the full posting, enough to touch on the most interesting and important highlights. And it's well written, and this is a perfect look into the process of exploitation and just a classic mistake.

So this guy started with a Netgear R7000. He says: "After a long day of hard research, it's fun to relax, kick back, and do something easy. While modern software development processes have vastly improved the quality of commercial software as compared to 10 to 15 years ago, consumer network devices have largely been left behind. Thus, when it's time for some quick fun and a nice confidence boost, I like to analyze Small Office/Home Office (SOHO) devices. This blog describes one such session of auditing the Netgear R7000 router, analyzing the resulting vulnerability, and the exploit development process that followed. The write-up and code for the vulnerability described in this blog post can be found in our NotQuite0DayFriday repository."

He says: "The first step when analyzing a SOHO device is to obtain the firmware. Thankfully, Netgear's support website hosts all the firmware for the R7000. The Netgear R7000 version 1.0.9.88 firmware used in this blog post can be downloaded from the website. After unzipping the firmware, we'll use binwalk to extract the root filesystem from the firmware image. While the router may have many services worth analyzing, the web server is often the most likely to contain vulnerabilities." And I'll add that it's also always listening with a port open to the LAN, so an obvious target of opportunity, if you could do anything useful on the LAN side.

So they continue: "In SOHO devices like the R7000, the web server must parse user input from the network and run complex CGI functions that use that input. Furthermore, the web server is written in C and has had very little testing, and thus it is often vulnerable to trivial memory corruption bugs. As such," he writes, "I decided to start by analyzing the web server HTTPD. As we're interested in how the web server mishandles user input, the logical place to begin analyzing the web server is the receive function. The receive" - abbreviated R-E-C-V - "is used to retrieve the user input from a connection.

"Thus by looking at the references to the receive function in the web server, we can see where the user input begins. The web server has two helper functions which call receive, one used in the HTTP parser and one used to read the responses from Dynamic DNS requests to oemdns.com."

He says: "We'll focus on the former use, as shown below in the Hex-Rays decompiler." And he has a snippet of decompiled code. Of course Hex-Rays, remember, is the IDA, the Interactive Disassembler, which is the industry's longest running and very popular way of decompiling something which is only available in binary.

Okay. So I'm going to go through this, but this won't take long, unfortunately. He says: "After the call to read content - the receiver helper function - the parser does some error checking, combines the received content with any previously received content, and then looks for the strings named 'mtenFWUpload'" - as in firmware upload - "and \r\n\r\n" - which of course is carriage return/line feed, carriage return/line feed. And that's, you know, two CRLFs is typically the way you end ASCII-based input. And he says: "...in the user input.

"If the user input contains these strings" - that is, that mtenFWUpload and the two CRLFs - "the rest of the user input after these strings is passed to the abCheckBoardID function. Grepping the firmware's root file system, we can see that the string mtenFWUpload is referenced from the files www/UPG_upgrade.htm" - in other words, firmware upgrade - "and also Modem_upgrade.htm," he says, "and thus we can conclude that this is part of the router's upgrade functionality."

Then in his blog posting he breaks to post "1996 Calling. They Want Their Vulnerability Back." He says, and here it is: "Following the user input, we next look at the abCheckBoardID function. This function, shown below, expects the user input to be the firmware file for the R7000. It parses the user input to validate the magic value (bytes 0-3), obtains the header size (bytes 4-7), and checksum (bytes 36-49); and then copies the header to a stack buffer. This copy, performed via the memcpy function, uses the size specified in the user input." All right, everybody. I'll say that again. The copy, performed via memcpy, uses the size specified in the user input. In other words...

Leo: Well, the user knows best. Of course.

Steve: ...it is trivial to overflow the stack buffer. Unbelievable.

Leo: Yeah.

Steve: He says: "In most modern software, this vulnerability would be unexploitable." First of all, having that vulnerability is unconscionable. The idea that you're going to take the input from the input is nuts. But that's what this does.

He says: "Modern software typically contains stack cookies which would prevent exploitation. However, the R7000 does not use stack cookies. In fact, of all the Netgear products which share a common codebase" - thus the reason that 79 of them all fall to this exploit - only, he writes, "the D8500 firmware version 1.0.3.29 and the R6300v2 firmware versions 1.0.4.12 through .20 use stack cookies. However, later versions of the D8500 and R6300v2 stopped using stack cookies, making this vulnerability once again exploitable against all of them, as well."

He says: "This is just one more example of how SOHO device security has fallen behind compared to other modern software." Then he says: "In addition to lacking stack cookies, the web server is also not compiled as a Position-Independent Executable, and thus cannot take advantage of ASLR." You know, Address Space Layout Randomization. "As such, it's trivial to find" - what he says, he writes - "a ROP gadget" - of course that's Return-Oriented Programming gadget, meaning the tail end of some existing routine that has a snippet of code that he wants to repurpose, he said - "within the HTTPD binary, such as the one shown below, that will call the system with a command taken from the overflown stack."

And so it's just two instructions: MOV RO,SP, meaning from register zero and the stack pointer, and then a branch if less, BL, to system. Meaning it is trivial to put your own command on the stack and have the system execute it. He says: "The exploit in GRIMM's NotQuite0DayFriday repository uses this gadget" - this Return-Oriented Programming gadget - "to start the telnet daemon as root listening on TCP port 8888 and not requiring a password to log in."

And: "As the vulnerability occurs before the Cross-Site Request Forgery (CSRF) token is checked, this exploit can be served via a CSRF attack. If a user with a vulnerable router" - that is, any of those 79 - "browses to a malicious website, that website could exploit the user's router via the browser." He said: "The developed exploit" - and it's all there - "demonstrates this ability by serving an HTML page which sends an AJAX request containing the exploit to the target device. However, as the CSRF web page cannot read any responses from the target server, it's not possible to remotely fingerprint the device." You could do other things to do that, like from the web page establish a communication with a web browser, get the version number and so forth.

Anyway, he says: "So the attacker must know the model and version that they are exploiting." These guys didn't want to build a full-blown, seriously deadly proof of concept because it would be immediately weaponized by bad guys.

They said: "Many SOHO devices share a common software base, especially among devices created by the same manufacturer. As such, a vulnerability in one device can normally be found in similar devices by the same manufacturer. In this case I," he writes, "was able to identify 79 different Netgear devices and 758 firmware images that included a vulnerable copy of the web server. This vulnerability affects firmware as early as 2007." That was the WGT624v4, version 2.0.6. "Given the large number of firmware images, manually finding the appropriate gadgets is infeasible. Rather, this is a good opportunity to automate gadget detection." And he goes on to do that. But anyway, everybody gets the idea.

He notes at the end of his post that on June 15th, so, what, Monday before last, Vietnam's ZDI group, the Zero Day Initiative, published an advisory by "d4rkn3ss" from VNPT ISC, that's the Vietnam ZDI, on this vulnerability. That's the other group that independently discovered this. Adam's GRIMM group discovered the issue independently and reported the vulnerability directly to Netgear on May 7th of this year. So, what, a little over like a month and a half ago.

So obviously anyone owning and using a Netgear router of any model should start checking in with Netgear for news of new firmware for their particular router. And through this I've been assuming that no one would be crazy enough to have enabled web-based remote administration on their router's WAN interface. It is an option. It is, thank goodness, disabled by default. So somebody would have to turn it on on purpose. If by any chance you have done that for some reason, immediately turn it off or update your firmware. But really, having a web browser on the WAN interface, as this exact problem demonstrates, is crazy. This is a zero-authentication attack on an extremely insecure web server. You do not want that facing the public Internet.

At the time of this podcast, Netgear had only addressed the problem for eight of their 79 vulnerable router models. Their advisory was first published last Thursday the 18th. And they have since updated it three times as they prepare and release additional firmware updates. I'm quite certain this is a nightmare for them, and they're doing everything as quickly as possible to get the firmware fixed and released. I mean, I found this page, that grc.sc/netgear, which of course is my shortcut that redirects to their much longer URL. It was the top item on their support advisories. So you can also just go to Netgear support and look at security advisories, and it's right there, "Security advisory for multiple vulnerabilities on some" - uh-huh - "routers, mobile routers, modems, gateways,

and extenders." So, I mean, it's across their product line. They just used the same bad web server everywhere.

I have a Netgear cable modem that I love, and an ASUS WiFi router. But the Netgear cable modem is outboard of my pfSense firewall router, and the ASUS is inboard, serving not as a router, only as a WiFi access point. So for me, having that professional-grade pfSense firewall provides a great deal of peace of mind. I can't recommend it highly enough. You can install on any little network-enabled gizmo - a Raspberry Pi, a little Intel fanless box, anything. Or just buy one of their little appliances for a couple hundred dollars. Anyway, it's great technology.

Meanwhile, DDoS is alive and well and growing. We haven't touched on DDoS attacks much recently because there hasn't been much news. But breaking a record is always newsworthy, and the record for the largest sustained DDoS attack was recently broken. The second highest previous record was set at a whopping 1.3 terabits per second - that's 1.3 trillion bits per second, 1.3 thousand billion bits per second - which hit GitHub back in February of 2018. That was topped a month later by the second, current - what had been the record holder since at 1.7 terabits per second, which was aimed at NetScout in March of 2018. Both NetScout and GitHub, both of those attacks abused Internet-exposed memcached servers to achieve their massive bandwidth.

But now, although they didn't advertise it at the time, in an incident quietly disclosed in its AWS Shield Threat Landscape report, Amazon's AWS service disclosed that they had successfully mitigated the largest ever DDoS recorded, weighing in at 2.3 terabits per second, 2.3 thousand billion bits per second. They did not disclose the intended target, but they did indicate that the attack was carried out using hijacked CLDAP servers, resulting in three days of elevated threat for its AWS Shield staff.

CLDAP, which is Connectionless Lightweight Directory Access Protocol, which is an alternative to the older LDAP protocol, which is used to search, connect, and modify Internet-shared directories, it's been abused for DDoS attacks since late 2016. And CLDAP servers are known to be able to amplify DDoS traffic by 56 to 70 times its initial size, making it a highly sought-after protocol, and also a common option provided and used by DDoS-for-hire services. Attacks this large, 2.3 terabits, are fortunately still quite rare, and may surprise those running attack mitigation services. There are far more much smaller attacks happening pretty much continuously.

Cloudflare, who mitigated a 550 gig per second attack during the first quarter of 2020, noted that 92% of all the DDoS attacks they mitigate, that is, Cloudflare, or mitigated during the same first quarter of 2020, were under 10 gigabits per second, 92. So in other words, only 8% topped 10 gigabits per second. And they also noted that 47% of all the attacks were even smaller, under 500 megabits per second. So it's very much a curve with a long tail and then a sharp exponentiation at the very end. Very few super high bandwidth attacks. Lots of much lower bandwidth attacks.

And as we know, DoS and DDoS are one of the consequences of the Internet's autonomous packet routing system which has served us so well from the start. As long as it's possible to query a remotely located public server with a UDP packet that does not require TCP's roundtrip, and if the query's source IP can be spoofed without being blocked on its way out onto the public Internet, and if the remote server's reply to the query is much larger than the query itself, amplified DDoS spoofing attacks are going to be a feature of our global Internet of networks until we eliminate some of those conditions for that kind of attack.

So a piece of errata. We don't often have much. I titled this "My, how time flies."

Leo: Uh-oh.

Steve: And as I mentioned at the top of the show, Leo, our off-the-cuff and inaccurate discussion of how long we've been at this podcast generated some conversation over in GRC's Security Now! newsgroup, the upshot of which was someone named Rob Allen summarizing our current position quite succinctly. Rob wrote: "We are currently in the 15th season, or Year 15, of the show, though it has only been 14 years and 10 months since August 18, 2005. On August 17th, the show will have been on the air for 15 full years, making August 11th the final episode of year/season 15. Season/year 16 will start August 18, 2020 with Episode 780." So we're at 772, so eight episodes from now we will be beginning Year 16 of this podcast. So there we have it.

Leo: Wow, wow, wow.

Steve: We've been almost 15 years. And the gang over in the newsgroup mentioned TimeAndDate.com, which is a pretty slick site for performing various sorts of time and date math. I did a quick computation about when our final podcast would occur.

Leo: Oh.

Steve: Yeah. I thought that would be interesting. So since we're at 772, and we run out of digits of course at 999, that's 227 remaining podcasts. So at one podcast per week, that's 1,589 days, which is 4.35 years. But since we'll have some holiday best-of, that'll extend that number by another four weeks, since we'll cross four holiday events. So it's really around 1,617 days from now. I dropped that number into TimeAndDate.com, launching from today, what is 1,617 days from now. And that places our final 999th podcast at Tuesday, November 26th, 2024.

Leo: Perfect.

Steve: Which really isn't that far away, you know.

Leo: No, shut up. Don't say that.

Steve: We're going to be there in no time.

Leo: Dammit, no. Can we go just to the end of the year? How about that? We'll do a holiday episode. Give me a couple extra. All right, all right, all right.

Steve: I don't know how we'll number them, Leo.

Leo: We don't have to give a number.

Steve: A, B?

Leo: Yeah, A, B, and C. Be 99A, B, and C. That settles it.

Steve: Actually, we could do -3, -2, -1, and zero.

Leo: We've never done the zero. We can't really do a podcast that doesn't start at zero.

Steve: Oh, we can do 0, 00, and 000 because I do have three digits.

Leo: You see?

Steve: So that could work. Yeah, I don't know.

Leo: It'll also be...

Steve: That does sort of fudge the ending. It makes it a soft...

Leo: It'll also be close to 20 years of TWiT, yeah. It's good, yeah. Well, it's up to you. I don't, you know, TWiT's going to reach its thousandth right about then, as well. So that will be intriguing, to see what happens after we go to four digits.

Steve: So I do have a piece of freeware to announce, which I think our listeners may find useful. It's Windows, even though it's got some DOS connection. It's called InitDisk, I-N-I-T-D-I-S-K. You could google "grc initdisk." It already turns up. Or it is under GRC's main menu under Freeware/Utilities. Or you can just go to GRC.com/initdisk. The back story is that to aid the forthcoming testing of SpinRite's new technologies, we needed a simple way to prepare a bootable USB drive. And because I always start from scratch at the beginning, from the bare metal, and then build up from there, we wound up with a uniquely capable new utility.

During our testing, a number of those who were testing found that USB thumb drives they had long believed to have died and to be dead were immediately brought back to life by InitDisk. And again, I wrote it all in assembler. It's from scratch. I wanted to basically take complete ownership, create a really robust formatting utility for removable USB. And just this morning I encountered another report by a frequent contributor who goes by "Obiwan" in the newsgroups. He had not been participating all along, so he wasn't aware of InitDisk's ability to bring dead USB drives back.

He said: "Tried it on a rather old TDK TF10 (8GB) USB stick which was 'dead,'" he has in quotes, "that is, Windows was unable to recognize it. At best it was recognized, but a very small capacity." He said: "Found it inside a closet. Not sure how long it was there. At any rate, downloaded" - meaning InitDisk - "and started InitDisk. Upon startup it asked me to insert the device," which is the way InitDisk operates, the way it identifies what you want to reformat. He says: "...asked me to insert the device. Did so by inserting the" - and again in quotes - "'dead' key. InitDisk recognized it," he said, "so I went on with the 'NUKE.'"

That's what you actually have to do. Once InitDisk recognizes it, it shows you everything it knows about it to help you identify it. And then you have to type N-U-K-E in order to give it permission. So he says: "...went on with NUKE and, at end, had the 8GB stick alive again." He says: "Not bad, I think. Further checks performed using various tools reported it to be okay. So, well, at least that InitDisk tool may be useful to recover, to some extent, not pretending miracles, some USB sticks."

So anyway, as I said, I don't think Obiwan had been following along, so he wasn't aware that we had seen a number of these instances. So the industry has a bit of new freeware from GRC. As I mentioned, it's the first of what I expect will be a couple offshoots from SpinRite's work. The next thing will be a very cool bare metal bootable mass storage device benchmarking tool. That's what I'm immediately going to start working on. Given the fact that GRC's DNS Benchmark remains our number one most downloaded utility, it's now at nearly 5.8 million downloads, and it's being downloaded at a rate of 3,000 per day, my hope is that this forthcoming mass storage benchmark will also develop a following.

And there's a method to my madness here. Everyone who uses it will also be simultaneously testing and verifying the new hardware driver suite that will be then moved into SpinRite. So the more early testing we can get of that new code, the better. And also, anyone wishing to verify that the next and all future SpinRites will work for them on whatever hardware they have will be able to use this free benchmark utility, which utilizes all of the same technology, to verify that for themselves, to make sure. And of course, if not, I want to know about it so I can fix it now. So I'll also be launching shortly a new set of GRC forums to make reporting and managing of any problems much easier for the users of the benchmark. So lots of happy activity going on over here, and some cool new code being produced.

You can optionally have InitDisk create a bootable FreeDOS disk for you. You just add the command "freedos" after "initdisk." It's not the default because most people are not going to want that. They're just going to want a nice reformatted USB stick or drive. You can even use one, like I used a USB-to-SATA interface, and it beautifully formatted. It'll run on any drive up to 2.2 terabytes, which is the limit for the 32-bit sector count which the MBR, the Master Boot Record, uses. This is not yet GPT. We will be at some point adding that, but not at this point yet. This is just to make it easy to boot things for people who want to play with my future forthcoming work in FreeDOS.

Okay. Ripples in the space-time continuum. The rhetorical question is, just how much damage can a little two-person company situated in Cincinnati, Ohio, do to the world? The answer to that question was likely revised this week. The company is Treck, T-R-E-C-K, Inc. They've got a very up-to-date-looking website at https://treck.com.

The site's home page declares: "Since 1997" - so 23 years - "Treck has been designing, distributing, and supporting real-time embedded Internet protocols for worldwide technology leaders." What could possibly go wrong? No, it doesn't say that. They said: "The Treck TCP/IP stack designer and Treck cofounder has more than 20 years experience and is a leading expert in embedded Internet protocols."

Well, as we know, anyone can be forgiven for making a mistake. And arguably, someone should probably have given a good hard look at these offerings before allowing 23 years of embedded device adoption to have occurred. But now is when we are. And as the Hacker News summed it up in their headline, "New Ripple20 Flaws Put Billions [with a B] of Internet-Connected Devices at Risk of Hacking."

The Hacker News wrote at the top of their coverage: "Dubbed 'Ripple20,' the set of 19 vulnerabilities resides in a low-level TCP/IP software library developed by Treck which, if weaponized, could let remote attackers gain complete control over targeted devices

without requiring any user interaction. According to Israeli cybersecurity company JSOF" - which I'll just be saying as JSOF - "who discovered these flaws, the affected devices are in use across various industries ranging from home consumer devices to medical, healthcare, data centers, enterprises, telecom, oil, gas, nuclear, transportation, and many others across critical infrastructure."

So switching to JSOF's summary and to get some additional details, they said: "The JSOF research lab has discovered a series of zero-day vulnerabilities in a widely used low-level TCP/IP software library developed by Treck, Inc. The 19 vulnerabilities, given the name Ripple20, affect hundreds of millions of devices or more and include multiple remote code execution vulnerabilities. The risks inherent in this situation are high. Just a few examples," they say. "Data could be stolen from a printer, an infusion pump's behavior could be changed, or industrial control devices could be made to malfunction. An attacker could hide malicious code within embedded devices for years. One of the vulnerabilities could enable entry from outside into the network boundaries. And this is only," they said, "a small taste of potential risks.

"The interesting thing about Ripple20," they wrote, "is the incredible extent of its impact, magnified by the supply chain factor. The widespread dissemination of the software library and its internal vulnerabilities was a natural consequence of the supply chain's ripple effect. A single vulnerable component, though it may be relatively small in and of itself, can ripple outward to impact a wide range of industries, applications, companies, and people. Ripple20 reached critical IoT devices from a wide range of fields, involving a diverse group of vendors. Affected vendors range from one-person boutique shops to Fortune 500 multinational corporations, including Cisco, HP, EMC, GE, Broadcom, NVIDIA, Schneider Electric, Intel, Rockwell Automation, Caterpillar, Baxter, as well as many other major international vendors suspected of being vulnerable in medical, transportation, industrial control, enterprise, energy (oil and gas), telecom, retail and commerce, and other industries."

They said: "Ripple20 is a set of vulnerabilities found on the Treck TCP/IP stack. Four of the Ripple20 vulnerabilities are rated critical, with CVSS scores over 9, and enabling remote code execution. One of the critical vulnerabilities is in the DNS protocol and may potentially be exploitable by a sophisticated hacker over the Internet, from outside the network boundaries, even on devices that are not connected to the Internet.

"A second whitepaper to be released" - because I looked at their first whitepaper - "to be released following Black Hat USA 2020" - so this is again a pre-Black Hat presentation announcement - "will be detailing the exploitation of CVE-2020-11901, a DNS vulnerability on a Schneider Electric ACP UPS device." They turn it off remotely with no authentication required. "The other 15 vulnerabilities are in ranging degrees of severity, with CVS scores ranging from 3.1 up to 8.2, and effects ranging from denial of service of the device to potential remote code execution.

"Most of the vulnerabilities are true zero-days, with four of them having been closed over the years as part of routine code changes, but remained open in some of the affected devices. Three were lower severity; one was higher. Many of the vulnerabilities have several variants due to the stack configurability" - that is, the TCP/IP stack configurability, which things were included and which were not - "and code changes over the years. Ripple20 are the only vulnerabilities reported, as far as we know" - in other words, nobody ever looked before - "except for some general logical vulnerabilities referenced in the past, which pertain to many stack implementations and usually had to do with RFC misinterpretations or deprecated RFCs." So just, you know, standard spec things.

They said: "Ripple20 vulnerabilities are unique both in their widespread effect and impact due to supply chain effect and being vulnerabilities allowing attackers to bypass NAT and

firewalls and take control of devices undetected with no user interaction required. This is due to the vulnerabilities being in a low-level TCP/IP stack" - meaning also in the kernel - "and the fact that for many of the vulnerabilities the packets sent are very similar to valid packets, or in some cases are completely valid." And in fact I looked at the first whitepaper where they fragment a DNS packet, and it's completely valid, but it causes a buffer overflow that allows you to inject your own code into any of these devices. "This enables," they wrote, "the attack to pass as legitimate traffic," because it is.

So to give some sense for what has just happened, I have the top six vulnerabilities with the highest severities rated from 10 down to 9. And of course 9 is still "house on fire." So the number one most severe vulnerability, the one I referenced, and this is a CVE-2020-11896, and this is from the formal CVE severity scoring. This is not these guys pumping this up. This has a severity score of 10.0, right, on a scale of one to 10: "Improper handling of length parameter inconsistency in IPv4 UDP component" - that's DNS - "when handling a packet sent by an unauthorized network attacker. This vulnerability may result in remote code execution." And I'll just tell everybody that you fragment a UDP, a DNS packet, and you get to run code on these devices.

Also 10.0: "Improper handling of length parameter inconsistency in IPv6 component when handling a packet sent by an unauthorized network attacker. This vulnerability may result in possible out-of-bounds write." So you get to write over areas that you're not supposed to. Again, executing code.

"9.8: Improper handling of length parameter." And so forth. Anyway, there's 19 of these. And anytime you see "improper handling of length parameter" it's very much like the problem we just talked about in the web server in the 79 Netgear firmware instances, where you're able to provide the length. It is the case that reassembly of fragmented packets is fraught with problems. I don't know why, but in the old days that was a constant source of errors. And these guys at Treck made the errors.

And unfortunately their code has been widely adopted. It's written in C. They advertise that it runs on everything, any microprocessor, any embedded platform, with or without an underlying operating system, on an RTOS or not. I mean, it's like everybody's answer. And it's probably affordable. So it got widely adopted. And now billions of devices have it. And as will be detailed at Black Hat, further detailed, are insecure.

JSOF attempted to be as responsible as possible. They said that their disclosure had been postponed twice as pleas for more time came from some of the many vendors that they were able to determine were affected. And of course they have limited surveillance abilities, with some of the vendors voicing COVID-19-related delays for the reason they weren't able to act more quickly. So out of consideration for those companies, the time period was extended from 90 to over 120 days. And even so, some of the participating companies became, they wrote, difficult to deal with as they made extra demands. And some, from their perspective, seemed much more concerned with their brand's image than with patching the vulnerabilities. In other words, they want it to be not mentioned as being, you know, like they were a company or their stuff was vulnerable.

So we know today that hundreds of millions, in the best case, of affected devices scattered all across the globe will almost certainly never receive security patch updates to address these critical Ripple20 vulnerabilities. We are truly in, probably looking into the future, a world of hurt. And in some despair, and not knowing what else to recommend, ICS-CERT recommended consumers and organizations, they had two bullet points: "Minimize network exposure for all control system devices and/or systems, and ensure that they are not accessible from the Internet." Unfortunately, their job may be to be accessible from the Internet. "Locate control systems networks and remote devices behind firewalls" - unfortunately that won't be effective - "and isolate them from the business network."

And then they said: "Besides this, it's also advised to use virtual private networks for securely connecting your devices to cloud-based services over the network." Which again may not be at all practical due to the nature, the embedded nature of these things.

So we talked last week about the dangers inherent in having a technological monoculture. At that point we were talking about Chromium and the Brave browser, I think. And of course they had switched to Chromium, too. Here we have a highly defective, very vulnerable TCP/IP networking library that has been on the market for more than 20 years, more than two decades. It is riddled with just-discovered critical vulnerabilities of varying degrees of severity from 10.0, many of them, down.

And there is just no way to update most of the affected devices. For those that are recently released, that are still in a maintenance contract or maintenance loop, maybe they've got self-updating facilities. But lots of embedded things don't. We have said till we're blue in the face how crucial it is that anything that is connecting these days have a means of updating itself because that's the only way to get the job done.

And most users who are not listening to this podcast will never even be aware that anything like this has happened. And even those of us who are privy to this podcast may not know what TCP/IP stack random embedded things have. Do our netcams have that? Maybe. Probably. Maybe not. Who knows? I mean, this is an internal library compiled into the ROM in these devices. They may not even be updatable at that level. And yet all of these networks and the network security we're all depending upon has arguably just taken probably a noticeable hit.

And you can absolutely bet that state-level and other hackers for hire are rubbing their hands together. And I'm sure that lists of affected hardware are being assembled at the moment. And it may be that people will just send test pings or test UDP, fragmented UDP at stuff and see if it crashes. If it does, it's probably got this Treck firmware down in its internals. The Internet is already a target-rich environment, and it just got a whole lot richer.

Leo: It's interesting that a number of these vulnerabilities are mishandling of length variables.

Steve: Yes.

Leo: Others are not sanitizing inputs. It's almost the same problem as with the Netgears.

Steve: Yup, classic.

Leo: How did this go undetected for so long?

Steve: Yeah. That's just it.

Leo: It's just not open source. That's part of the problem; right? So nobody...

Steve: Correct.

Leo: And it's embedded, so probably nobody was testing it.

Steve: Yeah. I think that it worked, I mean, and that's just it. Some engineer somewhere said, oh, look. It works, and I'm done. My problem has been solved because my boss said we need to hook this to the Internet. So, yay.

Leo: Oh, gosh. Golly. Just amazing. Wow.

Steve: Yeah.

Leo: All right, Steve. There we go, once again, you've made me all excited about the future of the Internet. This one's not going to get fixed. Period. Period.

Steve: No, it's not.

Leo: It's in embedded devices.

Steve: It's everywhere. It's too widespread. It's in our printers and in our light bulbs.

Leo: Wow. And you mentioned using a VPN to connect to our IoT devices.

Steve: No, that was just CERT saying...

Leo: I don't think that's a good idea, either.

Steve: Yeah, that's right. Put your light bulb on a VPN. That'll work.

Leo: No. All right. Oh. How depressing is this? What could you do? What would the mitigation be? Since we don't know which of our many, many, many devices with embedded TCP stacks are vulnerable.

Steve: The only practical thing you could do would be to create a purpose-specific filter like in a router. So your router would have to be trained about never accepting fragmented UDP. There have been other UDP fragmentation attacks. And you could argue that there isn't a good reason for UDP to be fragmented nearly as much as it once was. So you could just say "Drop any UDP packet with the frag bit set in its header." And so that would immediately protect you from some of these vulnerabilities.

And so you could argue, you could go through and carefully develop - and I imagine that some of the big IDS vendor guys are probably going to do that. They're going to add filters to protect their clients' internal networks from anybody sending this stuff from the outside. But as a device just raw, plugged into the Internet, I don't think there's any way to, I mean, it's just not going to get its firmware updated.

Leo: God. That's so depressing. So depressing. Well, thank you, Steve, for cheering me up, anyway. What are you going to do? It's good, I mean, we've got to know this stuff; right? That's why the disclosure happens.

Steve: Yup, forewarned.

Leo: Yeah. Steve Gibson's at GRC.com. That's where you'll find him and this show and 16Kb versions of the audio, which is wild. I don't know who's downloading those. Do people download those every week?

Steve: Yeah, they do.

Leo: You can check? Yeah? Wow.

Steve: Yeah.

Leo: Somebody wants them. They sound like Thomas Edison on his cylinder, his first recorder. But they're there. You can understand them. You can also get 64Kb audio, which sounds normal. You can also get, and this is maybe the most valuable version - well, as normal as we get. You can also get a beautifully crafted transcript by Elaine Farris, so you can read along as you listen. We have audio and video at our site, TWiT.tv/sn. By the way, when you're at GRC.com getting all that stuff, read all the other things there. It's fun.

And pick up a copy of SpinRite. If you get v6, you'll get v6.1 the minute it's available. You also get to participate in the beta tests. And you can already see, Steve's like NASA. All that research spins off into useful products. You know, it just spins off. So you'll be part of the spinoff. Just go to GRC.com and pick up SpinRite, and then you can spin off.

Our site is TWiT.tv/sn. It's also on YouTube. Best thing to do is subscribe in your favorite podcast application. You'll get it automatically - Overcast, Pocket Casts, Stitcher, Slacker, Podcast Republic. I can go on and on. You know what I'm talking about. Find the show. Subscribe. You'll get it every week, the minute it's available.

We do the show every Tuesday, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. Watch it live as we produce it, TWiT.tv/live, or listen live. And you can also chat, if you're listening live, at irc.twit.tv. After the fact, our TWiT community is open, as are Steve's forums. Steve's forums are at GRC.com. Ours is at TWiT.community. Thank you, Steve. Have a great week, and we'll see you next week on Security Now!.

Steve: Thanks, buddy. Right-o.