SECURITY NOW!

Transcript of Episode #777

rwxrwxrwx

Description: This week we revisit the trouble with F5 Networks' BIG-IP devices, we update on the epic Twitter hack, and we look at a security update for GnuTLS. We also cover the big five-day Garmin outage and Cisco's latest troubles. We'll point out a new Win10 debloater app and a bit of errata. Then I want to wrap up by sharing some truly surprising and interesting results that are emerging from my work on the pre-SpinRite hyper-accurate storage benchmark.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-777.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-777-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He's ready to go. Big trouble in BIG-IP, we'll talk about that. The Garmin ransomware breach, disk benchmarking, and another reason why you should be very afraid of GnuTLS. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 777, recorded Tuesday, July 28th, 2020: rwxrwxrwx.

It's time for Security Now!, the show where we protect you and your loved ones by setting your permissions high. This is the guy who puts the wall up, protects us all, Mr. Steve Gibson of GRC.com. Hey, Steve.

Steve Gibson: Hello, Leo. Great to be with you again. This is a landmark episode, 777.

Leo: Triple seven.

Steve: And we have a fun show title, too, for those geeky among us: rwxrwxrwx.

Leo: Now, let me ask you. When you use chmod, do you prefer chmod 777?

Steve: No.

Leo: Or do you prefer u+rwx?

Steve: Oh, I see, the style.

Leo: Do you use the letters?

Steve: No, I like to explicitly specify...

Leo: You like numbers.

Steve: ...exactly what I'm looking for, you know, like 640 is...

Leo: Yeah, because you know in your head what 640 is, that's why.

Steve: Exactly.

Leo: Yeah, yeah.

Steve: Exactly.

Leo: Occasionally, if I want to turn the execute bit on, I'll just do u+x.

Steve: Yeah.

Leo: But most of the time it's easier just to do the number. I agree with you.

Steve: So we've got an interesting episode. Nothing really stood out, and one thing that's interesting has happened in this past week is that some truly surprising and interesting results have been emerging from what has turned out to be a hyper-accurate storage benchmark that I'm working on for the technology for SpinRite. So there was some interesting - I just know that our listeners, odd as it sounds - and you, too, Leo. When you see this stuff, you're going to be going, what the what?

Anyway, we're going to start with security. We're going to look at the trouble that F5 Networks' BIG-IP devices are now having. We talked about this at the beginning of the month when it was a warning that, uh-oh, patch. We're going to update on the, what is it, I guess two weeks ago now epic Twitter hack. There's more information coming to light. We're going to look at an important security update for GnuTLS and those applications that are relying on it, and hopefully either have been patched or will be soon because there's a glaring problem.

We're going to cover the big five-day Garmin "outage," in quotes. And I'm going to break my promise - well, it wasn't a hard, set-in-stone promise - never to talk about ransomware again because when something significant happens, it's just I'm not talking about it every week anymore. And we're also going to talk about Cisco's latest troubles. I've had a pointer actually from BleepingComputer to a new Windows 10 debloater app, and we have a bit of errata.

Then, as I mentioned, I want to wrap up by sharing some of the really interesting results that are emerging as a result of the fact that I've ended up developing what is turning out to be what I would describe as a hyper-accurate storage benchmark. We're getting results accurate to four significant digits, and it's revealing some surprising things about our mass storage. So I think an interesting podcast for our listeners.

Leo: Yeah, you know, ransomware feels like it's gotten nastier and worse.

Steve: Oh, Leo.

Leo: Yeah. It just really feels that way, doesn't it. It's like we knew it would get worse, but yeah.

Steve: Yeah, it's time for us to talk about it. We need to touch in on it again because it really has.

Leo: I've got a picture.

Steve: We have a picture. And it's, first of all, I should describe it. We have a somewhat unhappy, shocked-looking computer user who's staring at the screen, which is announcing, "You have 10 updates." And it says, "6 slow your PC down. 3 look very dodgy. 1 randomly changes all your PC settings."

Leo: Must be using Windows 10.

Steve: Well, yeah. And so on one hand there's the cartoon itself. But what occurs to me is sort of the meta level, which is that some cartoonist created this cartoon because it's now, like, in the popular culture. I mean, it's enough of a problem that it's not...

Leo: Good point, yeah.

Steve: ...geek land talking about this.

Leo: It's not just us, yup.

Steve: Yeah. It's like, oh, I mean, like everybody knows this. It's like this is not going to be good for me. I mean, I have to do it, I guess. Well, actually now it's not a guess. Yes, you know, Microsoft will make you do it. But you do it, and random things happen, which aren't clearly in the customer's best interest. So anyway, I just - I thought it was interesting. Not only that yes, okay, sure, you know, funny cartoon. But the fact of the funny cartoon, you know, the fact that somebody is drawing a cartoon that says this, says wow, we're not really doing a service for our customers in the industry all the time.

So at the very end of last month, I think it was June 30th, F5 Networks released a critical patch for their so-called BIG-IP systems. It was a "maximum vulnerability" is the way it was termed, remote code execution flaw that they disclosed...

Leo: That's right. They don't get worse than this, baby.

Steve: ...in their so-called "TMUI," the Traffic Management User Interface of the BIG-IP - which is actually like a trademark, I don't know what it stands for, I mean, IP we know, Internet Protocol, but BIG, maybe it stands for something other just big - their application delivery controller, ADC. They go to initials here. Anyway, this came to light as a consequence of F5 publishing this patch. And with it was an urgent call for users of these so-called BIG-IP systems to immediately update with the highest possible priority. And F5's customers using these BIG-IP solutions are governments; Fortune 500 firms; banks; service providers; well-known brands including Microsoft, Oracle, and Facebook. I mean, this is big iron.

So as we noted at the time, F5's website boasts that 48 of the Fortune 50 rely on F5. So somehow they missed two of the top 50 companies in the U.S. And at the time of the disclosure, so not quite but almost a month ago, more than 8,000 of these BIG-IP F5 Networks devices were found online publicly accessible on the Internet and vulnerable to attacks designed to exploit this vulnerability. U.S. Cyber Command independently urged F5 customers to patch their devices urgently. They tweeted: "Patching CVE-2020-5902 and 5903 should not be postponed over the weekend. Remediate immediately."

Leo: Wow.

Steve: F5 also offered some interim mitigation measures that they recommended for their customers who could not for whatever reason patch their BIG-IP equipment immediately. You know, sometimes that requires you take it down for some length of time and reboot. But it later came to light that the mitigation could be mitigated and bypassed, which made emergency patching the only safe course, like do it now.

So two days after the patches for this critical vulnerability were released, researchers started publicly posting proof of concept exploits showing just how easy it would be to exploit them. So that was then. Three weeks later, last Friday the 24th, the Cyber and Infrastructure Security Agency (CISA) posted, they said: "CISA is issuing this alert in response to recently disclosed exploits that target F5 BIG-IP devices that are vulnerable to" blah blah blah.

"Unpatched F5 BIG-IP devices are an attractive target for malicious actors. Affected organizations that have not applied the patch to fix this critical remote code execution vulnerability risk an attacker exploiting that CVE to take control of their system. Note: F5's security advisory states that there is a high probability that any remaining unpatched devices are likely already compromised. CISA expects to see continued attacks exploiting unpatched F5 BIG-IP devices and strongly urges users and administrators to upgrade their software to the fixed versions. CISA also advises that administrators deploy the signature included in this alert to help them determine whether their systems have been compromised." And so the signature was a traffic inspection script in order to see whether there was bad stuff going on.

They said: "CISA has observed scanning and reconnaissance, as well as confirmed compromises, within a few days of F5's patch release of this vulnerability. As early as July 6th, CISA has seen broad scanning activity for the presence of this vulnerability across federal departments and agencies. This activity is currently occurring as of the publication of this alert." Meaning, okay, from as early as July 6th is when it began. And this alert was last Friday the 24th. So this has been going on.

They conclude: "CISA has been working with several entities across multiple sectors to investigate potential compromises relating to this vulnerability. CISA has confirmed two compromises and is continuing to investigate. CISA will update this alert with any additional actionable information."

Okay. So this is a classic example. And actually this sort of ties into where we'll be going here in a minute when we talk about Garmin. I've often been speaking about the growing critical need for companies, and to a lesser degree individuals, but certainly individuals who care, to be certain that they have and are maintaining an open channel of communication for receiving vulnerability notices. I've been talking about email as that channel. But in thinking about this further, I think that Twitter likely makes the most sense now.

As I noted last week, Twitter really has become our global information dissemination platform, warts and all, for better or for worse. No one imagines that the announcement of critically patched vulnerabilities won't immediately be public anyway. Typically these things are made public, and any company that tried to send out email to a large customer base thinking that it would not end up immediately coming to everyone's attention is nuts. So it ought to be broadcast. I'm sure the bad guys have signed up to receive vulnerability announcements in email from all of these providers anyway, and no one is making sure that an announcement like this is not going out. The CVEs are overtly public.

So it seems to me the way this needs to work is for technology companies, that is, who are producing these things, to create an authenticated vulnerability announcement Twitter account which never contains corporate promotional nonsense. It's in the companies' interest to keep that vulnerability announcement channel named specifically for that purpose and clean and dedicated to nothing but the disclosure of the availability of important updates to correct, you know, which are meant to correct important security vulnerabilities. And it should be seen as beneficial to its reputation that it's committed to getting this news out as quickly as possible, in a flash fashion.

You know, I mean, the one thing they could do would be to time it so that the areas where it's most likely to be affected are awake. So, you know, don't send it out at 2:00 a.m. That's not going to be good. Wait till maybe after morning coffee, give that a chance to take hold. And a company's record of its disclosure of these things over time, evidenced by its past feed of such things, should be a point of pride for the company and seen as an aspect of security for its prospective and ongoing customers.

And if an enterprise's entire IT security team were to subscribe to the security vulnerability Twitter feeds of the set of vendors whose hardware and software they're using, then even if one person missed it because they were in the car or driving, I mean, the point is, if it's a broadcast, it's not going to one person who got laid off last week and corporate IT forgot to keep looking at that person's corporate email for important alerts. Instead, everybody in the security team gets it. Somebody is going to escalate this thing and take action when they should.

It's just the model we're seeing now is that it is a race. And every month now, or week, we see instances where something important occurs. The bad guys are now, I mean, they're organized. They arguably are more organized than IT that's got other stuff on its plate than just doing security vulnerability remediation. They're trying to move other strategies forward. The bad guys have nothing but badness that they're looking at. So I just think it's really worth thinking about how, as somebody in IT, you can give your company an edge.

And on the flipside the companies that are producing this should not be sending this feed out in their standard PR, where the feed ends up being bogged down with all kinds of other stuff. That doesn't invite its attention. And a company producing a security alert on a Twitter account wants it to be an account for that purpose. That way on the receiving end the IT team can be subscribing to these things and know they're important and not have them just buried in a bunch of noise. So I just, you know, email, yes, that, too. But I just think it's important for announcements to get out, more now than ever.

And speaking of Twitter, it's obvious in retrospect that if high-profile accounts were compromised so that attackers were able to obtain login access, they would or could have also nosed around in the normally private DM channels of those accounts. It's not something that was talked about last week, but it has since come to light. That is, not only were those compromised accounts used for sending out that two-for-one bitcoin deal, which was crazy, but it did try to make some money. And as we know, the transfers of, what, $240,000 in bitcoin generated by that little brief campaign was blocked, to everyone's benefit. Twitter updated, I don't remember when it was. Oh, on the 22nd, which is Wednesday last week. They updated their original blog posting, adding this little bit of information.

They said: "We believe that for up to 36 of the 130 targeted accounts, the attackers accessed the DM inbox, including one elected official in the Netherlands."

Leo: Remember, they said eight at first.

Steve: Uh-huh. Right. Right. And they said: "To date, we have no indication that any other former or current elected official had their DMs accessed." So not surprisingly, the news that some of the world's most influential people probably had their personal messages read by hackers who are still unknown, at least publicly to us, will put additional pressure on Twitter to better protect its users. And U.S. Senator Ron Wyden - who's generally one of the more technically savvy politicians, you know, he's on top of these various issues of encryption and technology and so forth. He said that he has pushed Twitter's CEO, of course Jack Dorsey, to protect direct messages with end-to-end encryption.

Ron said: "Twitter DMs are still not encrypted, leaving them vulnerable to employees who abuse their internal access to the company's systems, and hackers who gain unauthorized access. If hackers gained access to users' DMs, this breach could have a breathtaking impact for years to come." And of course from Twitter's standpoint, it would be a big feather in its cap if it could boast true end-to-end encryption for private DMs.

The idea, of course, would be that neither Twitter nor anyone else except the tweet's intended recipient would be able to read the tweets. And thinking of it from a standpoint of a crypto challenge, I'd love to be given the job of designing that system since it represents a number of interesting challenges. But probably the best person anywhere would be Matthew Rosenfeld, whom we commonly refer to as Moxie Marlinspike.

Leo: I didn't know that was his real name.

Steve: Matthew, or rather Moxie, and his crypto team at Signal, would be best suited to designing end-to-end encryption for Twitter. I still recall how weirdly overdesigned the

Signal protocol appeared to me at first when I was digging into it for our in-depth episode on that. But that feeling morphed into deep technical appreciation once I saw that their crypto - what it was that their crypto ratchet and other mechanisms that they had created, what features and flexibility it enabled. And that's the sort of design expertise and inevitable crypto mistake sidestepping that Twitter needs. If anyone out there at Twitter is listening, and if you have any interest in following up on Ron Wyden's end-to-end encryption suggestion, please, please, please don't roll your own brand new ad hoc solution. I'll bet Moxie and his team would welcome a new and high-profile challenge.

And Leo, what has Twitter talked about with regard to security? Do you know if this is on their plate?

Leo: You know, I don't understand how encryption, I mean, it would only be for DMs, obviously.

Steve: Only DMs.

Leo: Because anything public there'd be no point.

Steve: And it would have to be per device. It would be tied to devices.

Leo: I don't know why they should, to be honest, since Signal exists. I think it's bending Twitter to do something different.

Steve: That it's really not suited for? So like nobody should receive or send, like, sensitive content over DMs?

Leo: No. Well, we know you shouldn't do that. Right now that's clearly a bad idea. And I hate to give people the impression that it would be safe to do so. I mean, of course, if they enabled the Signal protocol, that would be. But, you know, there's ways to enable it that maybe are less secure. You know, WhatsApp uses the Signal protocol. But does that mean that Facebook doesn't have the keys? I don't know if the keys are device-only. So it's an interesting question. I don't know. Yeah, I mean, if you're going to do it, that's the way to do it. You're absolutely right. I think it's a mistake to just say, look, we have a secure messaging system built within Twitter. It just doesn't seem like part of the mission.

Steve: Right. And it does feel like it doesn't fit with Twitter's inherently sort of open...

Leo: Right. It's public, yeah.

Steve: ...casual approach.

Leo: DMs to me on Twitter are properly used as a way to take a battle with another person private. Say, look, let's just handle this in DMs; or waving at somebody; or saying, hey, let's talk. I'm not sure it should be used for private communications.

Steve: Yeah. For me, I think, although I famously don't follow anyone, I know from looking at other people's feeds, they're following 1,300 people. And so for me a DM is a means of bringing something to someone's attention that I would like them not to miss. And in fact I know that's the way our listeners use DMs to me is for exactly that purpose. It's like, Steve, you know, this...

Leo: Well, they do that because you're not following anybody.

Steve: Right.

Leo: Because they could "at" you in a public way which is where most people do that on Twitter.

Steve: And I do watch that. I watch all of our listeners "atting" SGgrc, and so I sort of - I see those things go by.

Leo: Right.

Steve: But were I following hundreds of people who are tweeting, it's like it would be just easy to see it, you know, to lose it in the scroll.

Leo: That's true, yeah. Yeah, I mean, I guess there's no reason not to do it if it's implemented properly. You know, what Twitter's really trying to do, I mean, bottom line, besides the embarrassment of not getting hacked, is to make money. And they've had a hard time figuring out how to do that. Their advertising isn't working very well for them. They had a tough quarter again. Now they're looking at a subscription model. So maybe that would be a subscription feature. And then encrypted direct messaging.

Steve: Well, and we know that they gave up on SMS. So they had to give up on SMS. Used to be able to send a DM via SMS.

Leo: That would be inherently insecure.

Steve: But that says no client on the sender's side. So we can't do end to end that way.

Leo: Yeah, I mean, maybe they will do it. It's interesting, yeah.

Steve: So we've often referred to OpenSSL as the standard. But as we know, it's becoming quite long in the tooth. Professor of Cryptography Bill Buchanan recently summed up the situation with OpenSSL, writing in Medium, he said: "OpenSSL has caused so many problems in the industry, including the most severe with Heartbleed. The problem with it is that it has been cobbled together and maintained on a shoestring budget." And we've talked about this in the past. That's exactly the case. Many developers come and go. They're working on this or that extension to SSL, so they swing by the OpenSSL Project, graft on their new widget for live testing because it's like a great armature for that. Then they leave it hanging there without anyone to care for it moving forward. And it's really kind of amazing that it's done as well as it has.

As a consequence, today where there was once one, there are now many. Google forked OpenSSL to create - I love the name - BoringSSL. The OpenBSD project also forked it to create LibreSSL. And Leo, you and I talked about how Amazon took a different approach for securing the communications to their cloud services by creating a minimal subset of the whole named "s2n." And remember that that stands for "signal to noise" because the point was OpenSSL is so much code to do TLS that if you just said, let's start over, you could get a much better signal-to-noise ratio for your library.

And then just recently Google has created and released something we have not spoken about before, which is Tink, T-I-N-K, which is their new multilanguage cross-platform crypto library that can do TLS connections and gives applications access to that security. So there's all of those. And then there's GnuTLS, which is the subject here. It was first created a little over 17 years ago back in 2003. That happened as a means for allowing the GNU Project applications the ability to communicate securely over SSL and TLS.

Although OpenSSL existed at the time, OpenSSL's license was not compatible with the GPL. Therefore software licensed under the GPL, such as all of GNU's software, could not use OpenSSL without making a GPL linking exception. The GnuTLS library was licensed originally under the GNU Lesser General Public License version 2, which included applications using the GNU General Public License. Then in August of 2011, the library was updated to LGPLv3. But once it was noticed that there were new license compatibility problems introduced, especially with other free software, with the license change, in 2013 the license was returned to version 2.1.

So that's where we are now. One way or the other, under one license or another, GnuTLS has been around since 2003. And, not surprisingly, it has found its way into a great many applications. Just to get everyone's attention, I'll name a few: apt; cadaver, which is WebDAV, essentially; cURL; Wget; Git; GNOME; CenterIM; Exim; WeeChat; MariaDB; Mandos; Mutt; Wireshark; Rsyslog; slrn; Lynx; CUPS; gnoMint; GNU Emacs; Slapd; Samba; the Synology DiskStation Manager; OpenConnect; and a whole bunch of various VNC implementations. So, yeah, you know, it's the way you do TLS if you want something that's compatible with the GPL. And this is why it's worth taking note and looking into the situation more closely, when the result of the recent audit of GnuTLS is summed up with two words, "Be afraid."

Leo: Oh, boy.

Steve: It's not what you want...

Leo: No.

Steve: ...from the auditors of your TLS library. So Linux users need to determine how afraid they individually should be, if at all. Maybe the things they've had are already patched because this is a few weeks ago. NIST explains the problem very dryly by writing: "GnuTLS 3.6.x before 3.6.14 uses incorrect cryptography" - which is a nice way of putting it, as we'll see - "for encrypting a session ticket," they said, "a loss of confidentiality in TLS 1.2, and an authentication bypass in TLS 1.3." Neither of which you want in TLS, of course. "The earliest affected version is 3.6.4" - which was released September 24th of 2018, they said - "because of an error in a commit at 2018-09-18. Until the first key rotation" - after a connection to a TLS server based on this library - "it always uses wrong data in place of an encryption key derived from an application," based on the library.

So in other words, to make this a little less dry, when the news of these audit results broke, the cryptographer Filippo Valsorda, who's Google's security team lead for GO, their GO language, he tweeted: "Don't rely on GnuTLS, please." Then he says: "CVE-2020-13777. Whoops. For the past 10 releases, most TLS 1.0-1.2 connections could be passively decrypted," meaning all you need to do is capture the traffic. He says: "And most TLS 1.3 connections intercepted trivially." And he says, as an aside: "Also, TLS v1.2-1.0 session tickets are awful." But that's another issue.

So I have a link to the NIST announcement and the GnuTLS audit. Someone who's hip to security quoted Filippo's tweet, and he wrote: "You are reading this correctly. Supposedly encrypted TLS connections made with affected GnuTLS releases are vulnerable to a passive cleartext recovery attack, and active for 1.3," meaning active attacks for TLS 1.3. He says: "But who uses that anyway?" He says: "This is extremely bad. It's pretty close to just switching everyone to HTTP instead of HTTPS, more or less. I would have a lot more to say about the security of GnuTLS in particular, and security in general. But I am mostly concerned about patching holes in the roof right now," meaning how is he affected by this.

He says: "So this article is not about that." He says: "This article is about figuring out what exactly was exposed in our infrastructure because of this." So again, I have a link to his full coverage. There are some, for example, if you are using Debian, there are some commands you can use with the package manager to quickly discover which packages you have installed which link to any of the affected versions of GnuTLS, which would tell you what you currently have.

As I said, this is now a few weeks ago. So I would imagine that there's been a lot of GnuTLS updating and relinking and package reissuing and updates. So the real takeaway for this is just make sure that the Linux you're running has been recently checked for any libraries that use GnuTLS, and that they've been updated. And if you are a builder, if you're using it yourself, you want to make sure you are up to date with the latest because this was a really bad problem since I guess September of 2018, for all of the instances of GnuTLS that have been out there, a trivial plaintext attack on the initial cipher. So very embarrassing.

Oh, and a bit later, summing things up, he writes: "The impact of this vulnerability depends upon the affected packages and how they're used." He says: "It can range from 'Meh, someone knows I downloaded that Debian package yesterday' to 'Holy crap, my full disk encryption passwords are compromised. I need to re-encrypt all my drives.'" Because, for example, one of the things that relies on GnuTLS was - I mentioned it. What was the name of that? It's the disk encryption package, Mandos. Oh, and including "I need to change all LDAP and MySQL passwords," which could have been impacted, too. So anyway, just a heads-up for Linux users. GnuTLS has probably been there, and you want to make sure that it's been updated since then.

The Garmin outage. And I put "outage" in quotes because, okay, yeah. I mean, technically that's correct. A screenshot from Garmin.com anytime between last Wednesday, late last Wednesday, and probably Sunday, stated, red banner across the top: "We are currently experiencing an outage that affects Garmin.com and Garmin

Connect. This outage also affects our call centers, and we are currently unable to receive any calls, emails, or online chats. We are working to resolve this issue as quickly as possible and apologize for this inconvenience."

And then in the show notes I have "And Now." And we can see this is a picture of my browser where the first tab is not in focus. It says "Security Now! #777 - 07-28-2020." And the tab next to it is www.garmin.com, which is updated. It says: "We are happy to report that many of the systems and services affected by the recent outage, including Garmin Connect, are returning to operation. Some features still have temporary limitations while all the data is being processed." And by that we probably mean decrypted. "We'd like to thank all our customers for your patience and understanding. Click for more details."

So as I mentioned at the top of the show, after last year's overkill on coverage of ransomware, because it was, you know, it just took off last summer, as I mentioned, I promised to stop mentioning this scourge of the industry week after week. And I've been good since then, even though ransomware attacks form a constant background. It's like, yeah, okay, fine, you know. And it's true. If some random dentist in Hoboken needs to cancel his appointments because his office has been hit, if indeed he's still in business after the novel coronavirus hit, and I agree that there are more pressing matters for this podcast's attention.

However, when a high-profile, highly networked, Internet-connected and Internet-dependent giant like Garmin gets its servers encrypted and needs to go dark, well, that's worthy of a mention. The troubles began late Wednesday and early Thursday morning as customers reported being unable to use a variety of their services. This came as no surprise to Garmin, and they tweeted at the time. I grabbed their tweet for the show notes. Basically it amounts to the message that they put across the banner of their website, so I won't read it again.

This service failure left their millions of customers unable to connect their smart watches, their fitness trackers, and whatever other devices to Garmin's servers and network that provided location-based data in some cases to make them work. And although many within the industry suspected exactly what it turns out by all reports internal and external did happen, Garmin's post yesterday was the first the company provided, yesterday as in Monday the 27th. Garmin's post was the first that actually gave us an official notification of what caused the worldwide outage.

They said: "Garmin Ltd. was the victim of a cyberattack that encrypted some of our systems on July 23rd, 2020. As a result, many of our online services were interrupted, including website functions, customer support, customer facing applications, and company communications. We immediately began to assess the nature of the attack and started remediation. We have no indication that any customer data, including payment information from Garmin Pay, was accessed, lost, or stolen.

"Additionally, the functionality of Garmin products was not affected, other than the ability to access online services. Affected systems are being restored, and we expect to return to normal operation over the next few days. As our affected systems are restored, we expect some delays as the backlog of information is being processed. We're grateful to our customers' patience and understanding during this incident and look forward to continuing to provide the exceptional customer service and support that has been our hallmark and tradition."

So screenshots which appeared and other data posted by employees suggested the ransomware was a relatively new strain called WastedLocker. A person with direct knowledge of Garmin's response over the weekend confirmed that WastedLocker was indeed the ransomware used. The person spoke on condition of anonymity to discuss this confidential matter with the technical press. WastedLocker first came to public attention just 2.5 weeks ago, on July 10th, when Malwarebytes published what they called one of their "Threat Spotlight" profiles. I have a link to the entire Threat Spotlight in the show notes for anyone who wants the full details. In their Spotlight, Malwarebytes said that WastedLocker attacks are highly targeted against organizations chosen in advance. And what we're about to learn, exactly as you were saying, Leo, this represents a change in the terrain and the application and deployment of ransomware.

So they said: "...highly targeted against organizations chosen in advance. During the initial intrusion, the malware conducts a detailed analysis of active network defenses so that subsequent penetrations can better circumvent them." So this is no longer an opportunistic botnet spray looking for things to affect. This is different.

Pieter Arntz, a Malwarebytes researcher, wrote: "In general we can state that, if this gang has found an entrance into your network, it will be impossible to stop them from encrypting at least part of your files. The only thing that can help you salvage your files in such a case is if you have either rollback technology or a form of offline backups. With online or otherwise connected backups, you run the chance of your backup files being encrypted, as well, which makes the whole point of having them moot. Please note that the rollback techniques are reliant on the activity of the processes monitoring your systems, and the danger exists that these processes will be on the target list of the ransomware gang, meaning that these processes will be shut down once they gain access to your network."

So Malwarebytes' posting also notes: "WastedLocker is a new ransomware operated by a malware exploitation gang commonly known as" - and I'm not kidding - "Evil Corp, the same gang that is associated with the Dridex and BitPaymer malware. The attribution is not based on the malware variants, as WastedLocker is very different from BitPaymer. What was retained was the ability to add specific modules for different targets. The attacks performed using WastedLocker are highly targeted at very specific organizations. It is suspected," they wrote, "that during a first penetration attempt, an assessment of active defenses is made; and the next attempt will be specifically designed to circumvent the active security software and other perimeter protection which the initial foray found to be in use.

"This effort represents a new and clear escalation of the ransomware scourge. We're no longer looking at opportunistic attacks which ask for some fraction of a bitcoin. If reports are to be believed, including the U.S. Department of the Treasury, the bad guys are now highly organized Russian cybercriminal gangs. They're not screwing around.

"The name 'WastedLocker' is derived from the filename it creates, which includes an abbreviation of the victim's name and the string 'wasted.' For each encrypted file, the attackers create a separate file that contains the ransomware note. The ransom note has the same name as the associated file with the addition of '_info.' Once the WastedLocker gang have taken hold in a network, their demands typically range from 500,000 to $10 million." And as we know, Leo, sometimes even more. "So this is the new face of international cybercrime extortion. If hackers delete or steal a company's data, there's nothing to extort. But if hackers encrypt a corporation's data, they're able to dangle the carrot of the decryption key. It's diabolical."

Garmin's notice yesterday did not employ the terms "ransomware" or "WastedLocker," but the description "cyber attack that encrypted some of our systems," all but definitively confirms that ransomware of some sort was behind the outage. And we have disclosures from unnamed but presumably reliable Garmin insiders to further confirm it. We all want to know whether or not Garmin paid up or restored from backups; and, if they anted up, what did they pay?

Sky News, citing a number of unnamed security sources, reported that Garmin did obtain the decryption key. And that report matched what the person with direct knowledge told members of the tech press, as well.

Leo: So they paid. They paid.

Steve: Yup. Sky News said Garmin "did not directly make a payment to the hackers," but did not elaborate further. However, as we've discussed on the podcast before, there are now middleman agencies who negotiate on behalf of their ransomware victim clients. Payment may have been made through such an intermediary. Garmin's representatives declined to provide confirmation that the malware was WastedLocker and whether or not the company paid ransom. And you know there's no benefit to them elaborating. And in fact it might actually cause them some trouble.

On December 5th of last year the U.S. Department of Treasury officially sanctioned Russia's Evil Corp, citing a Russia-based cybercriminal group as being behind the Dridex malware. The U.S. Treasury Department's announcement started off by saying: "Today, the U.S. Treasury Department's Office of Foreign Assets Control (OFAC) took action against Evil Corp, the Russia-based cybercriminal organization responsible for the development and distribution of the Dridex malware. Evil Corp has used the Dridex malware to infect computers and harvest login credentials from hundreds of banks and financial institutions in over 40 countries, resulting in more than $100 million in theft."

And it goes on, but that's enough to give us a taste for it. So the U.S. Treasury's action could complicate Garmin's position with respect to Evil Corp. Presumably, if a company is sanctioned, U.S. businesses are no longer allowed to have any commerce with it, and I guess one could argue that this paying extortion is commerce. So anyway, today Garmin's services are now mostly back online. As we know, and as we've commented before, attacks are driven by motivation. And few things motivate like cold, hard cash.

Ransomware has emerged as an insidious but viable technique for the extraction of cash from those who have it and those who have been caught without adequate failsafe fallbacks in the event of such an intrusion. And as our listeners know, when ransomware first appeared, we covered it on the podcast, and our reaction was "Uh-oh." Because it was clear that in-place encryption, coupled with cybercurrency, enabled this significant new threat. And now to that we add tightly targeted attacks launched by international organized cybercrime groups. So staying current with security updates, keeping employees on guard against intrusion spoofs, which as we know is the way 90% of these intrusions begin, and maintaining offline backups in the case bad guys get in anyway is the order of the day. And thus ends our ransomware reminder wakeup call for 2020.

Leo: Geez. Oh, lord.

Steve: It's really a problem, Leo. I mean, we've talked about how porous security inherently is; that if somebody wants badly enough to get in, they can find a way. And it's clear that we know nothing about the way they got in at Garmin. We did discover ultimately how Sony was breached. It's typically somebody doing something they shouldn't on the inside. But, boy, is it expensive. And we saw government institutions last year, lots of school districts that are cash strapped and didn't have the money to invest in IT. I mean, it's expensive to create. And it's difficult, too.

Think about all the workstations that are spread throughout a company like Garmin, where something, some malware could get a foothold, then start looking around, map out the network, figure out where things are, laterally move within the network unseen until they figure out exactly what's going on. And then I'm wondering why they didn't wait until late Friday night, why the attack took place on a Wednesday night. It would seem to me that the weekend is more disruptive. But anyway, I don't want to give them any ideas. Just amazing.

Leo: Incredible.

Steve: So Cisco. Unfortunately, speaking of going where the money is and limiting ingress to high-value targets, we have the sad patching status of Cisco's most recent critical vulnerability within tasty enterprise-grade devices. And when I tell you that it's yet another directory path traversal mistake, everybody try not to roll your eyes. It is. Last Wednesday the 22nd, Cisco released their security advisory with a CVSS score of 7.5. That's seeming somewhat worse, or things are seeming somewhat worse than that today.

Cisco's advisory reads: "A vulnerability in the web services interface of Cisco Adaptive Security Application and Cisco Firepower Threat Defense software could allow an unauthenticated remote attacker to conduct directory traversal attacks and read sensitive files on a targeted system. The vulnerability," they wrote, "is due to a lack of proper input validation of URLs in HTTP requests, processed by an affected device. An attack could exploit this vulnerability by sending a crafted HTTP request containing directory traversal character sequences to an affected device. A successful exploit could allow the attacker to view arbitrary files within the web services file system on the targeted device.

"The web services file system is enabled when the affected device is configured with either WebVPN or AnyConnect features. This vulnerability cannot be used to obtain access to ASA or FTD system files or underlying OS files." They said: "Cisco has released software updates that address this vulnerability. There are no workarounds that address it." Then, in an update to this initial disclosure, they said: "Note: Cisco has become aware of the availability of public exploit code and active exploitation of the vulnerability that is described in this advisory. Cisco encourages customers with affected products to upgrade to a fixed release as soon as possible."

Well, over time, this podcast has compiled a few golden rules of cybersecurity. I may not have explicitly stated this one, but it clearly ranks among the most important. Web interfaces are dangerous. Don't use them. Oh, yeah, they're pretty, and they mean that the IT guys don't need to read yet another boring manual listing confusing-looking commands. No, a web interface means that you can just fire it up and poke around in the menus until you find the button you're looking for, then press it. Unfortunately, so too can the bad guys.

And one well-established golden rule, as we know, is that interpreters are incredibly difficult to make perfect. Yet perfection there is required because the job being performed by most interpreters asks them to interpret untrusted content. And the interpreter in any web server is right up there in complexity and exploitability with that of any multimedia codec that we've run across. So a web server is an interpreter, though it is inherently facing the public Internet, and it is inherently accepting untrusted content from anybody who wants to send an HTTP URL.

Well, it turns out that this one also has a directory traversal vulnerability, meaning that you're able to put in the URL, as we know, ../../../.. in order to back out of the root directory of the HTTP server, back to the actual root of the file system, and then move forward down a different branch of the directory that you're never supposed to, as a remote untrusted user on the Internet, get to.

So Mikhail Klyuchnikov of Positive Technologies, who was credited with independently reporting this flaw, along with Ahmed Aboul-Ela who's with RedForce, said this vulnerability is highly dangerous. The cause is a failure to sufficiently verify inputs. An attacker can send a specially crafted HTTP request to gain access to the file system, which stores data in RAM, the so-called RamFS.

So last week, at the time of the disclosure, no attacks were known to be underway, but Ahmed Aboul-Ela of RedForce released a proof of concept which demonstrated the vulnerability, and he's been tweeting up a storm of example ways to exploit the flaw ever since. The bad news is it's horrifically easy to exploit this problem - this is Cisco - and also horrifically trivial to find vulnerable targets.

Which brings us to the state of affairs as of today. It's not good. As the update to Cisco's vulnerability announcement noted, attacks are underway. Rapid7 jumped on this and took a look at what it meant. In their report from last Thursday, they noted: "Rapid7 encourages immediate patching of vulnerable ASA/FTD installations to prevent attackers from obtaining sensitive information from these devices which may then be used in targeted attacks." In other words, exactly what the Evil Corp in Russia cyber gang is looking for. Rapid7 said, echoing Cisco: "There are no workarounds that address this vulnerability." Meaning, you know, it's a core problem in the parser, the URL parser of the HTTP web server in these Cisco appliances. Rapid7 said: "Cisco has provided fixes for all supported versions," blah blah blah.

Rapid7's Project Sonar discovered just over 85,000, Leo, of these ASA/FTD devices. And 398 of them are spread across 17% of the Fortune 500. Since it is difficult, if not impossible, to legally fingerprint Cisco ASA/FTD versions remotely, in other words so as to determine what version they are running, Rapid7 Labs revisited what's known as the "uptime technique" described in a 2016 blog post for another Cisco ASA vulnerability a couple years ago, four years ago. That shows, using the uptime technique, it shows that only 10% of affected Cisco devices have been rebooted since the release of this patch. In other words, 10% of 85,000 vulnerable devices have been rebooted since the release of the patch.

In their note they said rebooting is a likely indicator that they've been patched, yet only 27 of the 398 that are detected within Fortune 500 companies appear to have been patched and rebooted. So again, it's not possible to say this too often. Nothing is more important than making sure you've got open lines of communication to the software and hardware vendors of the equipment you're using and to have somebody who's on, like, absolutely watching this stuff. This cannot go to some neglected email account that the IT team checks on every week. One week is no longer fast enough. It should be clear to everyone by now that a vulnerability is no longer a surprise exception. Just ask Microsoft on any Patch Tuesday. And I can imagine that Dynamic Update and Patch Management could become a job title. I wouldn't be surprised if it does.

One piece of miscellany, and I can't vouch for this. It's called Bloatbox. Debloating Windows 10 after installation and before getting down to any serious business has become something of, like, what any serious user needs to do. Every time I install Windows 10, and I've got a bunch of installations around now doing different things, stripping all of the junk off of it is really what you have to do before you sit down to get any serious work done. Over time, I've assembled a few tools to do this.

I know, Leo, you've talked about some PowerShell scripts. I have them, too. And basically they amount to some PowerShell commands with wildcards for "please remove everything." And then there are a few extra things because, for example, things like Connect, which is extra stubborn and needs a little extra coaching in order to get it to leave. But so far the available utilities for accomplishing these tasks have left me unimpressed. I'm hoping that this one will be different. It's not clear.

And again, I just wanted to put it on everyone's radar. It's a newly released open source tool called Bloatbox. It's up on GitHub. If you just google "Bloatbox," the first link is to it. I've not had the occasion to use it yet, so I'm not vouching for it. And one concern is that it might be digging a bit too deep. So don't tell your unsophisticated users about it. In the sample screenshot that you've got on the screen right now, Leo, I see options to remove various versions of Microsoft.NET.Native.Framework and the Fluent XAML Theme Editor and more. Those are things that probably ought to remain where they are. So I would advise caution and only remove things that are recognizable, that are in your face and are annoying you.

Still, Bloatbox might be worth a look. I'll be on the lookout for any Twitter feedback about it from any of our listeners who do check it out. And the next time I'm facing a Start Menu loaded with flippy animated tiles pushing Candy Crush Soda Saga, I will definitely give it a try myself, and I will report back what I find. You know, I've been tempted to do something to fix this, but I know that our listeners would rather have me continuing to work on getting SpinRite out the door.

I did have a piece of errata that I thought was interesting and definitely worth sharing from a David A. Wheeler. He tweeted, I guess he DM'd, he said: "Hi. You've been claiming in Security Now! that the CVE number after the year is in sequential order." He says: "That has not been true for a long time. There are too many CVEs for one organization to assign them all. So there are now many CVE Numbering Authorities known as CNAs, each of which is given a block of integers to assign. So it's no longer as simple as number after year indicates order or indicates the number of vulnerabilities this year."

So David, thank you for bringing that up. We know that we do have a phenomenal number and that the total count is going up rapidly. But good to know that, if you were pulling CVEs from somebody who hadn't issued many and got a low number block, that could be happening late in the year, even though the number was low. So thank you.

Leo: On we go, Mr. Steve Gibson.

Steve: So this benchmarking software has evolved into a surprisingly accurate measure of performance. It's a bit like having access to a high-resolution microscope, and as a result we've been discovering some very interesting and surprising things. I have tables in the show notes which our listeners will probably want to take a look at. You might want to stick them onscreen, that first one. That's the result of seven separate runs of the benchmark against a system containing a 10TB Seagate spinning drive and a half-a-terabyte Crucial SSD.

Now, remember that the earliest hard disk drives maintained either 17 sectors per track, which were MFM, or 26 sectors if they were RLL. That's why they got that 50% increase in density, by using run-length limited encoding. So that was the number of sectors per revolution. And they had the same number of sectors around the innermost cylinder as around the outermost. And we've all sort of seen the original pictures of a pie-slice hard drive, where the slices represent the sectors. But that meant that the bit density of the bits around the inner cylinder set the bit rate for the drive, that is, the maximum density for the drive, and that the same number of bits were more greatly spread out around the outer cylinder because of course the outer cylinder has a much greater circumference than the inner cylinder. That clearly wastes space.

So all of today's modern drives vary the number of sectors around the track, depending upon the track's length. And that varies with the track's position on the drive, of course. So as a consequence, a modern hard drive's data transfer rate will also vary with the position of the track on the drive. So this chart was the result of, as I said, okay, one, two, three, four, five, six, seven runs of this benchmark on a 10TB Seagate drive.

In order to get a sense for this, I recently added position dependence into the benchmark. But what was initially happening was I using a random position. And people were saying, hey, you know, I'm getting different results every time I run the benchmark. And I of course knew why. But I thought it would be interesting to have the benchmark take readings at different locations in the hard drive. And as we can see from the table above, the position where the benchmark is taken greatly affects the data transfer rate. I'm measuring at zero, at 25% into the drive, at 50% into the drive, 75%, and 100%. So basically five places at quarter spreads. And as we would predict, the actual data transfer rate drops off as we move in toward the inner cylinders.

In the case of this 10TB Seagate drive, the back of the drive runs at about 46% of the throughput compared to the front of the drive. And so this suggests that for a spinning drive, moving the most often-accessed data to the front can more than double the drive's actual throughput, compared with data located at the end of the drive. And notice something else in that table that I'm quite proud of, and that is the remarkable run-to-run repeatability of the benchmark's results. I mean, in the case of the 50% point, they're all 205.3 megabytes per second; the 75% all 170 megabytes per second, with one of them at 169.6. And at the 100% point, 112.7 megabytes per second, with two of them at 112.6. So basically four digits of accuracy from the benchmark, which is what, first of all, lets us believe the numbers, and also it becomes sensitive enough for us to see things that we otherwise would not have noted.

Which takes us to the second chart showing on that same system seven runs of the 512GB Crucial SSD. We see the same sort of intertest repeatability. Basically the seven successive tests are identical. But something that's really interesting that I first noted here and that a lot of the testers have been noting is that the front of the SSD is significantly slower than later in the SSD. What we think this shows is there is fatiguing occurring at the front of the SSD which is reducing its performance. We know that there is active write-leveling that goes on in order to swap regions around so that one region that is being written often doesn't die. Instead, the SSD controller is remapping the regions of the SSD transparently.

It's not something that there's any UI for at the interface to the drive. But what we believe this means is that this wear leveling is not global in nature. It's local in nature. So there is a limit to the reach of the leveling across the drive. I think the most consistent performers we've seen have been Samsung SSDs, you know, the high-end ones, the 560 and, no, I guess it's...

Leo: 850, 860, and 870, yeah.

Steve: Yeah.

Leo: EVOs.

Steve: Yeah, they really do seem to be doing a better job.

Leo: Good, because that's the ones I buy.

Steve: Yeah, yeah. And I just think I believe in Samsung's technology. And then in this last table is something else that we've seen which is really interesting. So we've got a super accurate throughput benchmark giving us four digits of accuracy. We had one of our testers who has a system with four identical 2TB Seagate drives run the benchmark three times. So he produced three sets of benchmarks for each drive. This table he rearranged them by drive, so it's three benchmarks for the first drive. And in fact in the table you can see the "P" column is the SATA port that the drive is on. The "S" is the SATA speed. So SATA III, SATA II.

And in fact we had one tester didn't know it, but he had a SATA III-capable SSD on a SATA II port that he'd never noticed. The benchmark showed him that this SATA III device was on a SATA II port. And in fact the next version will explicitly notify you if you have that kind of speed mismatch. He moved his SSD from SATA II to SATA III and more than doubled the measured throughput for that device. So that was a nice little benefit.

But look at these numbers in this last chart. We see the four groups of three for each of the drives. The retest of a given drive shows almost identical results. But these four identical drives are showing differing performances at the zero, the 25, the 50, the 75, and the 100% points. For example, one of them shows right at the front 166 at the zero. Another one is 174, another one is 167, and another one is 158. So they're the identical drive. The retest is the same, but the three drives differ. And they differ differently across their area.

So what could account for the precise performance of identical drives staying consistent for each drive, but differing from the others? What differs from one drive to the next? What has always differed from one drive to the next? The number and the location of physical surface defects. This benchmark is revealing the subtle transfer timing variations which result from physical sector remapping around the defects. The location and number of defects differs from one drive to the next. No two are going to be the same. But of course they remain fixed for any single drive. At the moment I'm performing the benchmark by taking 32 consecutive back-to-back 32MB transfers of 65536 sectors each, so that's 1GB. So I'm going a 1GB read at the beginning, a 1GB at the 25% point, 1GB at the 50, 1GB at the 75 and at the 100.

I've proven that I'm eliminating all intertransfer overhead. No revolutions are being lost between blocks. So I'm streaming data off the drive at its maximum theoretical performance. And of course I developed all this for SpinRite because this is what's going to make SpinRite 6.1 scream. And for the benchmark, I've achieved a timing resolution down to the hundreds of picoseconds of accuracy, which is how I'm able to get the actual throughput readings so just dead on and repeatable, run after run.

But I mentioned that I have an idea for an improvement. Because these timing irregularities have raised some interesting questions, by next week's podcast, although I promise not to take up so much time, I will have changed the test to 33 consecutive back-to-back transfers, adding one. And I plan to snapshot the exact instant where the interblock of the 32 interblocks occurs so that we can more granularly see how each of the 32MB transfers flow. And I'm going to use that first transfer, that 33rd in front, so that I can discard the first one. That way the benchmark will be able to eliminate any head seek time and rotational latencies from the start of each block, which I'm not eliminating now. So that way the benchmark won't start timing until the first block is discarded, and that 32MB has been read.

So as I've mentioned before, the idea of creating a mass storage benchmark like this started out kind of as a bit of a Trojan horse, you know, an inducement for our listeners to obtain some value in return for their effort of formatting and booting a USB stick to run the benchmark on their various hardware systems. In the process of doing that, they would be verifying for me that SpinRite's new suite of bare metal, no BIOS drivers, of which this AHCI driver is the last and final that I need to develop, are working for them, and thus proving that SpinRite will work for them, too. But it's looking like this hyper-accurate storage benchmark is going to wind up providing some very interesting information for its users. So my original plan for a companion web forum was to help in managing any problems that people had with the benchmark, but I think we're going to also need a place to discuss people's interesting findings as they use this. So anyway, I just thought our listeners would find that interesting.

Leo: Cool.

Steve: Very, very cool timing results.

Leo: Yeah. And a very cool show. Thank you, Steve. We do Spin the Security Now! Bottle every Tuesday, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. If you want to tune in, you can watch us make the show live at TWiT.tv/live. There's audio and video streams there. Steve has copies of the show, though. You can always download those at his website. He's got 16Kb audio, 64Kb audio, and transcripts, a really nice feature. That's all at GRC.com. While you're there, pick up SpinRite, the world's best hard drive maintenance and recovery utility. Keep up on the updates on SpinRite 6.1 as he works on it. Participate if you want. In fact, if you pick up 6.0 now, you can participate in the development of 6.1 and be part of the team. That's all at GRC.com, along with a lot of great freebies, as well.

On-demand versions of the show at our website, as well, TWiT.tv/sn. We've got 64Kb audio and video. We also put it up on YouTube. You can watch it there, if you want, in a variety of formats to fit the device you're watching. And of course if you have a podcast application, the easiest thing to do would be just subscribe. That way you get it automatically the minute it's available every Tuesday afternoon. Steve, we'll see you back here for 778.

Steve: And you know, Leo, over the course - we're coming in here on the end of Year 15. That happens next month.

Leo: Yes.

Steve: And it occurred to me that over the course of the last 15 years the world has changed a lot, and there may actually be a reduced need for 16Kb audio now.

Leo: Does anybody download it? You must have numbers.

Steve: Yeah, oh, yeah, yeah. We do get downloads. I actually bounce them through - I guess I'm still bouncing them through Podtrac because you guys are...

Leo: Oh, okay. No, we don't use them anymore.

Steve: I think the links still work.

Leo: Actually, well, that's an interesting point. We have new redirects. I don't suppose anybody - Patrick, if you're listening, make sure you get Steve the redirects. We don't use Podtrac anymore.

Steve: Yeah. All of the high resolution actually just go to you. I'm using the same link you guys use.

Leo: It's just for the 16, huh.

Steve: Yeah.

Leo: Oh, okay, okay. Well, actually I'd be really curious to see. Probably, probably doesn't affect the results.

Steve: Probably not a big demographic, no.

Leo: Yeah. We'll send you the new redirects because we have different redirects these days. Thank you, Steve.

Steve: And just so all of our Linux listeners know, I'm not recommending rwxrwxrwx.

Leo: Never.

Steve: That is not a good idea. Yeah, that's not the way you want to leave your...

Leo: The only time I ever do that 777 is if I'm so frustrated and just go, chmod -R 777. Do it to everything. And then I can figure out after the fact. Actually, there's some programs, it's interesting, GPG, GNU Privacy Guard, which is the open source PGP, will complain - I think SSH will, as well - if it has [crosstalk] sessions on folders and files, which I think is really great. Yeah, yeah, that's a nice feature. Thank you, Steve. We'll see you next time on Security Now!.

Steve: Okay, buddy. Bye.