
# SHAmbles

**Description:** This week we look at some surprising revelations of Apple's cloud storage encryption (or lack thereof). We also cover a Microsoft cloud database mistake, some interesting legislation under consideration in New York, new attacks against a consumer router firmware, a rise of new attacks against our browsers, a welcome new publication from NIST on Privacy, a massive leakage of telnet usernames and passwords, a welcome micropatch for this month's IE zero-day, a bit of miscellany and SpinRite news, and then some coverage of the final nail that was recently pounded into SHA-1's coffin.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-751.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-751-lq.mp3

---

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We knew it was coming. It's been a long time coming. But finally it looks like SHA-1 is really deprecated. Steve explains why and what the future holds. We'll also talk about Apple's very confusing encryption policy. Is your data on iCloud safe or not, and what can you do about it? And yet another router problem from Tomato. Are you using Tomato? We'll talk about the Tomato router, open telnet ports, Microsoft data leaks, and a whole lot more. It's all coming up next at Security Now!.

**Leo Laporte:** This is Security Now! Episode 751, recorded Tuesday, January 28th, 2020: SHAmbles.

It's time for Security Now!. Oh, yes, he's here, ladies and gentlemen. I feel like we should have, I don't know, a big red curtain, and you could come out from behind the curtain, and the crowd would go crazy, on their feet. Johnny Carson style, you know? Mr. Steve Gibson, the head of research at GRC.com.

**Steve Gibson:** Well, we did that at the beginning of the SQRL...

**Leo:** Yeah, that was fun.

**Steve:** That was very nice. It was staged, but it was effective.

**Leo:** Yeah. Shhh, don't tell anybody.

**Steve:** So today we're finally going to get to talk about the subject that I've mentioned pushing off for the last several weeks because something more urgent kept coming up. And this, I can't take credit for this clever word. This is the word from the paper that was written, "SHAmbles," shambles, which is to say that our, well, I was going to say our much-beloved SHA-1 hash, but no one loves it anymore.

**Leo:** Oh, nobody loves it. Nobody.

**Steve:** Nobody loves it.

**Leo:** So let me - well, we'll talk about it. When we get to it, I have lots of questions to ask you.

**Steve:** Cool. But we do have other news, of course. We're going to take a look at some surprising revelations about Apple's cloud storage encryption, or lack thereof. We also cover a Microsoft cloud database mistake, some interesting legislation under consideration in the state of New York, new attacks against a consumer router firmware, a rise of new attacks against our browsers, a welcome new publication from the U.S. National Institute of Standards and Technology (NIST) about privacy, a massive leakage of telnet usernames and passwords and their IPs, a welcome micropatch for this previous Patch Tuesday/IE zero-day, a bit of miscellany, a little bit of SpinRite news - and of course SpinRite news, we're going to be having a lot more of that here coming up. And then some coverage, as I mentioned, of the final nail that was recently pounded into SHA-1's coffin. So I think another great podcast for our listeners.

**Leo:** Did you get the email from Intel this week?

**Steve:** No.

**Leo:** I got - actually it was from Digital Ocean. Yet another kind of - I was thinking it's speculative execution because it affects people on shared processors.

**Steve:** No.

**Leo:** But this one from Digital Ocean saying, yeah, we're going to have to fix this one, too. It just never stops.

**Steve:** Well, and so far we've not actually seen...

**Leo:** There's been no exploits.

**Steve:** ...any exploit, exactly. So, but on the other hand, we know that, if it isn't fixed, it will be exploited. And the bad guys will go to any extremes to make that happen. So, yeah, as long as it, I mean, it doesn't really affect our listeners very much, and I like to

mostly focus, well, accept where theoretical weirdness happens, which is fun to talk about, like what happened with speculation two years ago.

**Leo:** This came yesterday. "Hi there. Today Intel released a statement regarding two processor data leakage security vulnerabilities, vector register sampling and L1D eviction sampling, that may allow unintended information disclosure for users of multitenant cloud environments like Digital Ocean. Theoretically they could use a droplet to infer partial data used by previously run software or another droplet" - that's what DO calls the "instances" on this same post.

**Steve:** Boy.

**Leo:** So I don't know, are you familiar with these? Or are these new ones?

**Steve:** No, but I know what they just said.

**Leo:** You're way ahead of me.

**Steve:** What they just said was that the nature of caching is now under attack. And of course.

**Leo:** Of course.

**Steve:** What is a cache except a storage of recently used and probably likely to be soon reused information. And of course the reason we have caching is that the processors, the performance of the processors have so far exceeded the performance of main memory that you just can't afford not to cache it locally. And so that's what the L1 eviction probing or sampling is, is that they're saying there's a way for you to tell what's in your L1 cache based on the length of time it takes. And so, yeah, of course that's going to happen. So, I mean, I guess the one...

**Leo:** Kind of an amazing hack, isn't it, though?

**Steve:** Yeah.

**Leo:** Because it's not that they get the data. They get it from the timing.

**Steve:** Yup.

**Leo:** What?

**Steve:** And so it's an inference hack. There is on the horizon, we know, that there is that next-generation non - well, no, it is volatile. Is it volatile? I don't remember. I don't think

it's volatile actually, that high-speed crosswire memory technology. There is like the promise of finally solving the DRAM problem. This is all a fault of the fact that we need so much density in RAM now that a bit is now a tiny little capacitor. And you've got to refresh it. And the only way of sensing it is by dumping that capacitor into a sense amplifier which makes it a destructive read, which means you then have to write it back because, I mean, you can't just lose your memory the moment you query the memory. There are old people for whom that happens, but that's not our computers.

**Leo:** Me.

**Steve:** And so we're in this mess because nobody has figured out how to make that faster. When we make the capacitors too small, then we become susceptible to Rowhammer attacks because the noise immunity drops such that adjacent rows of reads affect the bits that we're trying to not have affected. So there's that problem. So, I mean, it's a mess right now.

And I think probably in the future we're going to end up with some next-generation technology that just doesn't have this problem. And when we can move memory onto the chip, which will probably happen, when we can move main memory onto the chip, then this whole need for L1, L2, and L3 caching potentially goes away so that we're no longer leaving a footprint of what we've done because at this point we have no choice. We have to cache.

And so the only solution is to cache per core, and then make sure that you don't switch a core outside of a virtual machine because that would then make it susceptible to probing from the other virtual machine. So, yeah, we're in a mess.

**Leo:** So this is not a speculative execution. This is a cache problem.

**Steve:** Right, right.

**Leo:** Well, there you have it.

**Steve:** It's a microarchitectural flaw.

**Leo:** It shows how hard this stuff is.

**Steve:** It's inherent in caching, yes.

**Leo:** And really it's not - I want to say it again, it's not that Intel made any obvious blunders.

**Steve:** This has been there from the beginning.

**Leo:** It's that the hackers have become incredibly sophisticated in the techniques. And it's not even hackers. It's really researchers at universities.

**Steve:** Yes.

**Leo:** Who've become incredibly sophisticated. These timing attacks like Rowhammer are crazy sophisticated.

**Steve:** Well, and it also sort of demonstrates what happens when you point academics at a whole new area. Two years ago somebody said, ooh, look what happens with branch prediction. And it's like, oh, well, if that's true, then this and this and this and this and this and this and this, I mean, like pretty much everything just collapsed.

So our Picture of the Week is fun. It's one that someone tweeted me, and I said, oh, my goodness, that's just a kick. And so I had a slot for it. We show sort of a large office environment with a bunch of desks and two in the foreground, the rest in the background. And one gal on the left is saying apparently to the guy on the right, she says: "I've asked the hackers to turn the thermostat down after they're done accessing our system." So, yes.

**Leo:** Love it.

**Steve:** The world we live in today. Speaking of the world we live in today, the question has arisen, and apparently it's been answered, actually, whether Apple is actually encrypting our iCloud storage backups. I wanted to start out this week by discussing this since we were talking about this recently. I was first made aware of it by a guy named Jeff Root, who is a frequent participant in our newsgroups, and I told him I'd give him a little shout-out, so there you go, Jeff. He pointed me at a story in Reuters which has since been picked up by a bunch of the tech press because it makes a little - after a lot of interviewing of former Apple employees and some FBI people who are in the know, they end up citing six sources familiar with the fact that Apple explicitly dropped its plan for encrypting backups after the FBI complained.

So these six sources told Reuters that Apple had dropped its plans to give iPhone users the option to fully encrypt backups of their devices that are backed up to iCloud, of course, after the FBI complained that the Apple's plan would harm their investigations. This happened about two years ago and had not been previously explicitly reported. Reuters stated that it showed how much Apple has been willing to help U.S. law enforcement. Of course we were talking about this back and forth over the last couple weeks as a result of this new Pensacola shooting.

But in addition to this, Reuters says that behind the scenes Apple has been providing the U.S. Federal Bureau of Investigation with rather comprehensive help, that is, as much as they've been able to, not specifically related to a couple of the more high-profile cases that we've talked about, but just sort of in general, citing one current and three former FBI officials, and one current and one former Apple employee.

Reuters reported that more than two years ago Apple told the FBI that it planned to offer users end-to-end encryption - and we take that now to mean nobody can see into it, that even they can't - when storing their phone data on iCloud. Under that plan, primarily designed to thwart hackers, Apple themselves would no longer have a key to unlock the encrypted data, meaning it would not be able to turn over material to authorities in a readable form, even under court order. Subsequently, during private talks with Apple, representatives of the FBI cybercrime agents and its operational technology division

objected to the plan, arguing it would deny them the most effective means for gaining evidence against suspects using iPhones.

And when Apple later spoke privately to the FBI about its work on phone security about a year later, the end-to-end encryption plan had been dropped, according to those six well-placed sources. Reuters said that it was unable to determine exactly why Apple had dropped their sweeping encryption plan, but that it had been. Another former Apple employee said he was told, without any specific mention of why the plan was dropped, or if the FBI was a factor in the decision: "Legal killed it, for reasons you can imagine."

**Leo:** That is important. "Imagine" is perhaps what that employee was also doing because there's a little dispute over this. Rene Ritchie, we talked about it a couple of weeks ago.

**Steve:** Oh, okay, good. I'm glad.

**Leo:** Rene Ritchie said he remembers that time, and Apple made a conscious decision, and this is actually applicable for this show, to instead of fail secure, to fail safe; that they were worried, if they did do Trust No One encryption, and somebody lost their password, yes, it would fail secure, but they wouldn't be able to help that person. So they opted - and that may be what Legal said is, no, we don't want to be on the hook for losing people's data. We'll make it fail safe. The password can be reset. Which means of course it's not Trust No One.

**Steve:** Well, and that is what Tim Cook has officially said was their position, that is, they decided, exactly as you said, Leo, that it would be a problem for their users if there was no recourse for the recovery of storage from phone data backed up to the iCloud.

**Leo:** And that's actually an important point because, if any company can say, oh, you forgot your password, we'll reset it, then it's not Trust No One; right?

**Steve:** That's absolutely right. And in fact what's a little bit slimy here is what Apple says. For example, because after reading all this I went back to their privacy stuff for iCloud just to figure out what it was they were saying that was different than this because, I mean, to give our listeners a sense for this, during the first half of 2019, so the first half of last year, which is the period covered by Apple's most recent semi-annual transparency report on requests for data it received from government agencies, U.S. authorities armed with court orders asked for and obtained full device backups and other iCloud content for 1,568 cases, covering about 6,000 accounts. And it turns out that U.S. Secret Service, or I guess secret U.S. intelligence court directives that it received over the same period of time, it responded to more than 18,000 accounts during the first half of last year. So Apple is busy disclosing iCloud backup data.

So anyway, I went over and looked. So they have a section for iCloud, end-to-end encrypted data. And what they've said is that, or what Tim has said is that the most sensitive data is end-to-end encrypted. And I thought, okay, what does that mean? So they said: "End-to-end encryption provides the highest level of data security. Your data is protected with a key derived from information unique to your device, combined with your device passcode, which only you know. No one else can access or read this data." So that all sounds, like, really good.

And then they said: "These features and their data are transmitted and stored in iCloud using end-to-end encryption: home data, health data, iCloud Keychain, payment information, QuickType Keyboard learned vocabulary, Screen Time, Siri information, Wi-Fi passwords." And then they said: "To access your data on a new device, you might have to enter the passcode for an existing or former device." So they enumerate carefully the things they do end-to-end encrypt, that is, which they have no visibility into, which I guess is meant to say, but everything else that we back up, we're not end-to-end encrypting.

**Leo:** Right.

**Steve:** And then they said: "Messages in iCloud uses end-to-end encryption." They said: "If you have iCloud Backup turned on, your backup includes a copy of the key protecting your Messages." And this is what's a little slippery. "This ensures you can recover Messages if you lose access to iCloud Keychain and your trusted devices." They said: "When you turn off iCloud Backup, a new key is generated on your device to protect future Messages and isn't stored by Apple."

So this is really slippery because they're saying Messages in iCloud also uses end-to-end encryption; but they're also saying, "and the key to decrypt them is stored there, too." And then I looked under their privacy, Apple.com/privacy, and they said: "Messages are only seen by who you send them to. Apple can't read your iMessages while they're being sent..."

**Leo:** In transit.

**Steve:** Uh-huh, "...between you and the person you're texting."

**Leo:** Right.

**Steve:** And it's like, okay. So, yeah, you know, our listeners know how to hear that. And what that says is, right, we send them to the iCloud before they're encrypted and after they're decrypted. Yes, we encrypt them in the iCloud, but we store their key alongside the encrypted Messages. Oh, because if you lost access to that, you wouldn't want to lose your valuable Messages. It's like, okay.

**Leo:** So we talked a long time with Rene, and he's promised to make a video on this because this is extremely confusing. There are Trust No One ways to store Messages, as well as passwords and certain other data. The default is of course not Trust No One, it's fail safe.

**Steve:** Yup.

**Leo:** But there are ways for - and this confused the hell out of me.

**Steve:** Well, I would call it "fail open."

**Leo:** Fail open.

**Steve:** Fail safe is being too nice. It doesn't fail closed, it fails open.

**Leo:** So there is, on Messages, a setting to do cloud Messages. And if you turn that on, Rene tells me, you've just turned off encryption storage. However, if you just have it be part of iCloud backup, it is encrypted. Now, maybe they're storing the key - remember they said the key has to be then parsed with your code, your unlock code, to decrypt. So maybe they're storing that partial key. His sense, and I think it's extremely confusing, so I'm not going to say this is absolutely the case.

I'll tell you what is absolutely the case, according to Apple. If you're using cloud backup for anything, you should assume, because it's not clear, even though Apple says what's on your iPhone stays in your iPhone, you should assume that Apple has access to that. And that means, by the way, not only law enforcement has access to it, but a sufficiently sophisticated hacker taking advantage of a flaw in iCloud or social engineering might also get access to it.

If you back up your Apple device, your iDevice, iPhone or iPad, directly to a computer, you check the box that says "encrypted backup" in iTunes, or I guess it's now - I don't know what it's called now, but in that app, and you give it a password, that is Trust No One at this point. Only you have access to that. That is fully encrypted. Apple cannot get it. Soon as you put it on the cloud, some is, some isn't, it's not immediately clear what is, what isn't. So from now on, for my backups, if I want real privacy, back it up to your computer locally, turn on encryption, use a password, and don't lose it because Apple can't fix it if you lose it.

**Steve:** Well, and I'd have to say, I mean, so many people say to me, because they think I know, oh, my iCloud backup is full. What do I do? And so my point is that everybody wants to have their precious little...

**Leo:** It's convenient.

**Steve:** ...phone backed up to the iCloud. Yeah, it's like, oh, look, it's magic.

**Leo:** It's more than convenient. Apple pushes it. They push it hard because they make money on it because it's full, you have to buy more. And they push it. They really strongly encourage it. And they, I think, should be blamed a little bit for giving the misimpression that it is fully secure. What's on your iPhone stays on your iPhone. That was the big ad last year in CES.

**Steve:** Yes.

**Leo:** By the way, the hubris of that was immediately handled bad karma because I think there was a crack right after that ad was up. So it wasn't strictly true. It isn't true. But there are ways to do it. I hope - I don't know if Rene has made that video yet. He's got to make that video because it isn't clear to anybody what is safe and what isn't safe.

**Steve:** And props to Rene if he can figure this out.

**Leo:** Yeah. I can't.

**Steve:** This is a mess.

**Leo:** Yeah.

**Steve:** And, you know, confusion, unfortunately, causes most people to go, well, oh, it's probably okay.

**Leo:** Well, the thing that pulled me up short is he said, yeah, if you turn on messages in the cloud, you are now turning off Trust No One Encryption. And I went, what? What?

**Steve:** Yeah. Well, and Android Central picked up on this. Their title was "Apple may have ditched encrypted backups; Google hasn't." And Android Central said: "A bombshell report from Reuters suggests Apple ditched end-to-end encryption for iCloud backups at the behest of the FBI. Citing several former Apple employees and FBI officials, the publication [Reuters] notes that Apple planned to switch to end-to-end encryption for iCloud putting it on the same level as iPhones and iPads but reversed" - meaning native local encryption - "but reversed course after consulting with the FBI." So, yeah. That would be good if, I mean, I think Rene would have himself a very popular piece of video if he were able to unscramble this.

**Leo:** Yeah, because no one can translate this.

**Steve:** No, it's just nuts. Microsoft, to change companies, had a little slipup. On December 5th of last year, so late last year, Microsoft misconfigured the access controls of a database of more than a quarter billion, so 250 million, technical support service records dating back 14 years to 2005. In some of the press coverage of this, someone said, "If you've had any contact with Microsoft Support in the last 14 years, your data was published."

So although they claimed that automated tools had removed unneeded personally identifiable information, Bob Diachenko, who we've spoken of before - he's the guy who's been making it his personal crusade to go find exposed databases on the Internet, he's a database sleuth - he spotted the unprotected database, reported it to Microsoft, and said there remained ample readable information, including email addresses, IP addresses, locations, descriptions of claims and cases, Microsoft Support agent email, case numbers, resolutions, and remarks. Oh, and internal notes marked "confidential."

So anyway, I'm not sure when he found it, but he knows that it appeared on the 5th of December. On New Year's Eve, I thought it was interesting, they were kept late. They were there on December 31st, turning this off. So it disappeared from public exposure as a consequence of this misconfigured access control on New Year's Eve. But it was up there for three weeks or so. Not a big deal. But I just didn't want to skip over it completely without making note of the fact that that had happened, in case anyone cared.

Okay. So here's an interesting piece of news. New York state is exploring the possibility of introducing legislation to ban the use of taxpayer funds for paying ransoms, ransomware ransoms.

**Leo:** Oh, interesting.

**Steve:** Yeah. The state of New York is exploring whether outlawing the payment of ransomware ransoms might make the problem go away because the bad guys will know nobody in New York by law is able to pay. I'm not hopeful that that strategy's going to work. Two legislative bills have been proposed that would require government agencies to tell ransomware attackers they cannot pay.

The first bill, S7246, was proposed by Senator Phil Boyle on the 14th of January, obviously, what, exactly two weeks ago today. If it were enacted into law, it would for small cities and towns with populations under a million people restrict the paying of attackers with taxpayer money. If it were passed, it would set up a $5 million fund to help overhaul the IT infrastructures of such small towns. But 5 million, that would not go very far, unfortunately, under the terms of how much IT stuff costs these days. So, okay.

The second bill, S7289, introduced by Senator David Carlucci two days later, on the 16th of January, would prohibit government agencies from paying ransom in the event of a cyberattack against their critical infrastructures. At this point, the bills are both under discussion in committee, and it's unclear which, if either, of the bills would make it to a vote in the state senate. And of course we talked about last summer how in June of 2019 that U.S. Conference of Mayors passed a nonbinding resolution to sort of formally say we don't want municipal systems to be put back online as a consequence of paying ransom. We want you to say no. But of course their resolution was not binding. It was just like, okay...

**Leo:** They sort of said please, we beg of you.

**Steve:** Yeah, exactly. Maybe it gave mayors some cover to say, well, we don't want to pay, and the U.S. Conference of Mayors is behind us on this. But of course it lacked any legal teeth. So this New York legislation would be the first state to move in that direction, to like formally say it's against the law for a state agency, a state municipality to pay a ransom. So anyway, we'll see what happens.

That second bill, S7289, referred to the attack in Albany, New York last month - actually it was on Christmas Day - which paralyzed the Albany International Airport. At that time the attackers demanded a ransom in exchange for the return of the data and the restoration of the airport's systems. Because they were desperate, they paid a ransom. They're claiming, although the amount was not disclosed, that it was less than six figures.

So anyway, in response, the legislation says, in the bill, we don't want to keep rewarding these crooks for these attacks. They said: "When municipal corporations and government agencies comply with these ransoms, they incentivize cyberattackers looking to make a quick buck. Prohibiting these entities from complying with ransom requests will remove this incentive and safeguard taxpayer dollars." Uh-huh.

**Leo:** You know where they would be even more effective is if they also helped these agencies not get bit in the first place.

**Steve:** Yeah.

**Leo:** Establish some standards. Get some IT money in there.

**Steve:** Yeah.

**Leo:** It's one thing if they don't pay. Help them not have to pay; right?

**Steve:** Yeah. And Leo, in every case the municipality that gets attacked says, well...

**Leo:** We don't have a choice.

**Steve:** Yes. Give us the money. We don't have a budget that allows us to do - because, you know, in every case they have an IT infrastructure which is old and creaky. It's like it's barely working. Everybody's overworked and underpaid. And it's like, we'd love to be ransomware proof. But to do that, we've asked our IT people, what would it cost? And they quote something which we can't pay. So, yikes. It'll be interesting to see. I mean, I don't think this is going to work. But we'll find out.

There is a bad new botnet, the Muhstik (M-U-H-S-T-I-K) botnet, which is attacking Tomato routers. And we've not talked about Tomato, but it is a popular alternative router firmware. Palo Alto Networks Unit 42 researchers observed a variant of the wormlike botnet, this Muhstik botnet, that has added scanner technology to brute force web authentication against Tomato routers - and when I read this, I'm thinking, what? - on port 8080, which bypasses the admin web authentication using the default credentials. The defaults in the case of Tomato routers are admin and admin, or root and admin.

**Leo:** I would take root and admin. Go for that one.

**Steve:** Yes. The researchers wrote that they, quote, "captured Tomato router web authentication brute forcing traffic." Okay, now, I would take exception to the characterization of using admin:admin and root:admin as "brute force." I mean, okay, maybe wimp force.

**Leo:** I pushed it, and it fell over.

**Steve:** Exactly. Well, we gave it our first guess, and oh, look. We're in. We brute forced it. No.

Okay. So Tomato firmware, it's Linux-based, non-proprietary firmware which is known apparently not for its security, but for its stability. So VPN passthrough capability and advanced quality of service control. Yes, it's so advanced, even the hackers can specify what quality of service they want. It's typically used - apparently, multiple router vendors actually ship Tomato-based routers. I didn't know that, that there are some commercial vendors that said, oh, we're not going to bother creating our own firmware. Let's just use Tomato. It looks pretty. And in fact there is, there's Advanced Tomato.

So Tomato says of themselves: "Tomato is a small, lean, open source alternative firmware for Broadcom-based routers. It features a new user-friendly GUI, a new bandwidth usage monitor, more advanced quality of service and access restrictions, new wireless features such as WDS and wireless client modes, a higher peer-to-peer maximum connections limit, the ability to run custom scripts, connect via" - oh, this is wonderful - "telnet and ssh" - yes, telnet - "reprogram the SES/AOSS button, perform wireless site survey, and more." And then Advanced Tomato. I won't go into all the details, but basically it's really pretty. They said, you know, basically tired of those creaky old router UIs? Well, you want Advanced Tomato.

Anyway, so the point is that this thing has telnet, or at least web admin, rather, not telnet, web admin listening on port 8080. And as our listeners know, I'm a fan of the FreeBSD-based pfSense firewall router, so I don't have any firsthand knowledge of Tomato router configuration. But you and I, Leo, we've all heard of Tomato routers, and I'm sure our listeners have. It is inconceivable to me that any modern router could or would have its admin access enabled on its WAN interface.

**Leo:** Yeah.

**Steve:** I mean, it just - it can't.

**Leo:** Not by default, anyway.

**Steve:** No. Yes, exactly. Can you say RDP? But even assuming that the WAN interface binding must be manually enabled, in this day and age it is malpractice for firmware to allow any default login for the WAN-facing side. I'm encountering more and more software, and I'm always happy when I see this, which refuses to do things like that. If you are turning on WAN-side, it won't let you use a default credential. It won't have it. It'll say - and make you use a strong password and do a little bit of password strength metering and make you protect yourself. I mean, what year is this? It's crazy that you could simply click "Turn on WAN admin," and the default credentials which presumably haven't been changed facing the LAN, are now also facing the WAN. That's the only way you get admin:admin as your user:password, or root:admin.

So I just - I'm stunned that it's even possible. But of course that's the only way that the strategy of adding it to so-called brute forcing, although it's first guess in, of some wormlike botnet makes any sense is that, you know, it's like, oh, look, admin:admin. What do you know? Because, I mean, no user would choose that for themselves. They would do something else. Sarah or Wilbur or something. But not admin:admin. So it has to be that that's the default, and that people are turning it on, and then not changing the default username and password.

So, you know, if there's anyone in charge of router firmware within the sound of this podcast, please, I mean, how hard is it to refuse to not require the user to manually enter, make up some username and password when you, I mean, when you enable WAN-side admin. Which, by the way, is really bad. I mean, it's just bad.

**Leo:** Well, and who uses telnet anymore? I mean, SSH; right?

**Steve:** Yeah, well, actually we'll be talking about a little story about that here in a minute. But, yeah, exactly. SSH is what everyone should be using.

**Leo:** Telnet's in the clear. So even if you didn't have access, you could still see the traffic. So it just seems like a bad idea all around.

**Steve:** Yup.

**Leo:** Yeah. Now, I don't know, last time I checked Tomato hadn't been updated in years. I don't know if there's...

**Steve:** Yeah, there actually has been an update. I did take a look, and it was current. There have been some recent changes.

**Leo:** So we usually recommend DD-WRT. But they're related, I think. I feel like they're...

**Steve:** Yeah, there's DD-WRT, OpenWrt. There's something called Gargoyle because I was wondering about third-party firmware. Tomato. There's the LEDE project and then libreCMC are apparently the top six third-party firmwares. But I agree, DD-WRT and OpenWrt are the main ones that we talk about.

**Leo:** Apparently there are a lot of forks. And ASUS uses Tomato as the base of its entire line of home routers. I thought they were using DD-WRT.

**Steve:** That's interesting. It certainly doesn't look, I mean, it looks ASUS-ized.

**Leo:** Fresh Tomato is the only fork under "active."

**Steve:** Fresh Tomato.

**Leo:** It just begs to, you know...

**Steve:** You don't want to use the Rotten Tomato firmware.

**Leo:** No, use the Fresh, yeah. I wonder. That seems so weird that that would be turned on, WAN administration would be turned on, and then telnet port would be open.

**Steve:** Well, no, no. I misspoke there. We do have a story coming up.

**Leo:** Oh, this is not telnet. This is just the administrator WAN.

**Steve:** Yeah. This is port 8080 in order for it to open a port in the high port numbers.

**Leo:** All right. Yeah, there are a lot of Tomato forks. There's EasyTomato. There's Toastman. There's Shibby. Tomato ND, Tomato VPN, Tomato USB. There's a lot of Tomatoes. Lot of them out there. Wow.

**Steve:** Well, and Broadcom is a popular chipset. So it must be any hardware that is Broadcom based. It's like, oh, yeah, we've got your Tomato right here.

**Leo:** And I would hope any manufacturer shipping Tomato or a version of Tomato as their default firmware would fix that little flaw before they ship it.

**Steve:** I should hope.

**Leo:** You'd think so.

**Steve:** You'd hope. So our web browsers are under attack. Get a load of this one, Leo. The Chrome Web Store has been experiencing a wave of fraudulent transactions and has temporarily suspended publishing and updating of paid Chrome extensions due to a spike in fraudulent transactions. The Google security team has indefinitely suspended the publishing or updating of any commercial Chrome extensions on the official Chrome Web Store following a spike in the number of paid extensions engaging in fraudulent transactions. So let me repeat that. "The Google security team has indefinitely suspended the publishing or updating of any commercial Chrome extensions." So I guess only non-commercial extensions can be modified, just due to fraud.

Google said that the wave of fraudulent transactions began earlier this month. Google engineers described the fraudulent transactions as happening "at scale," meaning they couldn't deal with it. And this ban on publishing or updating impacts all paid extensions including Chrome extensions that require paying a fee before installing, extensions that work based on monthly subscriptions, or Chrome extensions that use one-time in-app purchases to get access to various features. Any existing commercial extensions that are already in place are still available for download via the official Chrome Web Store, but their developers are now forbidden from pushing any updates for fear that they might be fraudulent.

Simeon Vincent, who's the developer advocate for Chrome Extensions at Google, said: "This is a temporary measure meant to stem this influx as we look for long-term solutions to address the broader pattern of abuse." So it's apparently just gone out of control, and they just had to terminate any changes to commercial extensions.

Extension developers who try to publish a new paid Chrome extension or push a new update of their existing commercial extensions are currently receiving an automated message that all it says is "Spam and Placement in the Store." I saw a tweet, someone tweeted the message they got, which is terse and doesn't really explain what's going on. And since it's a blanket ban, even big name extensions have been impacted by it, the password manager Dashlane and meeting planner app Comeet.

The decision to ban publishing or updating Chrome extensions was formally announced just late last Friday night, on January 24th. Jeff Johnson, the creator of the StopTheMadness Chrome extension, told ZDNet that Google has been silently blocking updates for paid Chrome extensions actually before the official announcement, for a number of days beforehand. It's unclear how long the ban will last. Vincent said: "We're

working to resolve this as quickly as possible, but we do not have a resolution timeline at the moment." And he said: "Apologies for the inconvenience."

So, yikes. Wow. And, you know, it just sort of, when you think about it, browsers are now the way we interact with more and more of the world, the Internet, and browser-based apps are a thing. And it would make sense that extensions would come under attack.

**Leo:** But it's also part of this world where, if it can be gamed, it will be gamed.

**Steve:** Yes, yes.

**Leo:** It's just frustrating. People are so evil.

**Steve:** I know. It really is. I mean, it is despoiling the Internet.

**Leo:** This is why we can't have nice things.

**Steve:** Yeah. Also, and it's not just Chrome. Over the past two weeks Mozilla has banned nearly 200 Firefox add-ons in a crackdown on Firefox browser add-on misbehavior. Now, in this case, this wasn't a recent explosion. This is Mozilla basically making good on their promise to crack down on misbehaving add-ons and in ways that our listeners will find familiar because we've talked about this already.

So in total, Mozilla's add-on review team banned 197 Firefox add-ons which were caught with various forms of misbehavior - executing malicious code, stealing user data, using obfuscation to hide their code. The add-ons have been banned and removed from the Mozilla add-on portal to prevent new installs, and they've been retroactively disabled in the browsers of users who already had them installed. So 129 out of the 197 add-ons were all developed - that's a huge number, 129 - by a single developer named 2Ring, which calls itself a provider of business-to-business software.

And get this. In those 129 cases, the ban was triggered because 2Ring's add-ons were all found to be downloading and executing code from a remote server. And as we've discussed in the past, for obvious security reasons, Mozilla's updated add-on rule - and updated some time ago, so everyone had plenty of time to fix this - forbid add-ons from obtaining external code. And as we know, even if such code was benign now, there's no telling or controlling what any such code might do in the future. And if add-ons existed that did this, they would represent a very tempting target for supply chain attacks, which we've seen before, where somebody compromises the 2Ring server in order to insert malicious code that then these 129 different add-ons, whatever the hell they are, download and run.

So Mozilla has started strictly enforcing this rule across its entire add-on ecosystem. A different developer had six add-ons doing the same thing, and another one had three add-ons banned which were offering fake premium products. There were also bans levied against add-ons found to be improperly collecting user data: WeatherPool, Your Social, Pdfviewer, RoliTrade, and Rolimons Plus. There were also 30 add-ons banned for various types of malicious behavior. In those cases, only the add-on IDs were published so the developers could appeal the ban after curing the misbehavior. So there was also

questionable behavior spotted from something called FromDocToPDF add-on, and they found that it was loading remote content into Firefox's New Page tab.

So anyway, what we're seeing is a growing instance of browser add-on exploitation. And exactly as you said, Leo, it's sort of like, well, of course it's going to happen because there's a possibility for it. The idea is, of course, we have browsers. We want to create a richer browser behavior. We're all fans of add-ons - LastPass, famously. UBlock Origin, of course. Once upon a time NoScript. It makes sense to allow users to customize their browser the way they want to.

**Leo:** Well, especially if you're using a Chromebook because that's really the primary way you add features to a Chromebook.

**Steve:** Yes.

**Leo:** Is with Chrome extensions.

**Steve:** Yeah.

**Leo:** Wow.

**Steve:** Yeah. So we've got something of a mess on our hands.

**Leo:** Hard to fix this, yeah. I don't know what the solution is.

**Steve:** Well, the problem is anything that's done will create hurdles and restrictions.

**Leo:** Yeah. You could do what Apple's doing with their App Store; but people don't like that, either.

**Steve:** Right. And it's very much like the move to HTTPS. You could say, well, HTTP still works. You don't have to have a certificate. Except now the browsers are labeling any HTTP site as not secure. Which most end users are like, oh no, you know, like what does that mean? It's like, oh, it's just the way it used to be. And maybe the site doesn't have to be secure. But now you get dinged if you're not. So you can imagine if we do something to secure add-ons, it's going to make it more difficult, it has to make it more difficult for certifying an add-on developer somehow and making them obtain credentials and sign their work and so forth. It's like, okay. You know? It's just more difficult.

**Leo:** Yup. I think that's what's going to happen, though. That's the only way out. You have to do that.

**Steve:** Yeah.

**Leo:** Apple's proven it can work.

**Steve:** Yeah.

**Leo:** Apple's doubling down on that on the Mac, you know, they're going to require people to get notarized. You have to not only be a known developer, you have to be notarized to get an app running on the Mac.

**Steve:** Wow.

**Leo:** As problematic as I find that, it's also the right answer for this particular problem.

**Steve:** Yeah.

**Leo:** This malware issue.

**Steve:** So as we know, the U.S. federal government's National Institute of Standards and Technology (NIST) has produced several standards and guides over the years to aid organizations. Back in 2016 it published a set of password rules which, you know, we made some fun of from time to time. They were updated and improved. NIST also publishes a cybersecurity framework that has become a useful litmus test for those who need to secure their data.

NIST's latest such work is a new privacy framework laid out in a 43-page document aimed at helping to protect Internet users' personal privacy. I think it's a good thing. I read through it and looked at it yesterday. So this is a - and so it's NIST.gov/privacy-framework. The framework can be used when developing new products and services to ensure that that new work hasn't overlooked something important. So like a valuable checklist, just to sort of step through everything and make sure that you've got all your ducks in a row. Very much sort of like a pass, like "Is this password safe" set of rules.

This is more extensive. It can be used with an organization of any size. And I like the idea that you would look through, knowing what your company is doing, sort of go through this framework step by step and just sort of, I mean, even if you don't choose to make a change, at least this brought up the issue so you can be aware. And if you were to be able to be completely compliant, you'd be able to say we are compliant with NIST's formally established privacy framework.

**Leo:** This is good. I'm going to download this and send it to my executive team. I think we should take a look at this. That's good.

**Steve:** Yeah, I really do think it represents a nice step forward. In the executive summary, the framework describes itself, saying: "For more than two decades, the Internet and associated information technologies have driven unprecedented innovation, economic value, and improvement in social services. Many of these benefits are fueled by data about individuals that flow through a complex ecosystem. As a result, individuals may not be able to understand the potential consequences for their privacy as they

interact with systems, products, and services. At the same time, organizations may not realize the full extent of these consequences for individuals, for society, or for their enterprises, which can affect their brands, their bottom lines, and their future prospects for growth.

"Following a transparent, consensus-based process, including both private and public stakeholders, to produce this voluntary tool" - all this is is just a checklist - "the National Institute of Standards and Technology (NIST) is publishing this Privacy Framework: 'A Tool for Improving Privacy through Enterprise Risk Management' to enable better privacy engineering practices that support privacy-by-design concepts and help organizations protect individuals' privacy.

"The Privacy Framework can support organizations in" - and they have three bullet points - "building customers' trust by supporting ethical decision-making in product and service design or deployment that optimizes beneficial uses of data while minimizing adverse consequences for individuals' privacy and society as a whole; two, fulfilling current compliance obligations, as well as future-proofing products and services to meet these obligations in a changing technological and policy environment; and, three, facilitating communication about privacy practices with individuals, business partners, assessors, and regulators."

Anyway, it goes on and on and on at length. And I looked through it. It is very comprehensive. It sort of takes a tree branching approach, but ends up with a comprehensive bullet-point checklist. And I would suggest that anybody could take advantage of it. That is, you sit down and step through it with your IT group and just say, okay, where are we on this? Where are we on this? Where are we on this? And, you know, you could certainly make a conscious decision not to do that, or to say, oh, yeah, we probably need to be better about that. And it's like, okay. Let's write that down and talk about it next month sort of deal.

So anyway, I wanted to bring it to the attention of our listeners since I think it would be - and you obviously see this, too, Leo - an extremely useful tool, just to create a series of discussion points, rather than just sort of saying, without any focus, well, are we protecting our users' privacy? And it's like, okay, well, here it is. I mean, it is comprehensive. So anyway, I commend our listeners to take a look at it. NIST.gov/privacy-framework.

**Leo:** Yeah, it's good.

**Steve:** And here's where I got confused about telnet. Get a load of this. A hacker has leaked more than half a million, more than 500,000 telnet credentials for IoT devices.

**Leo:** Half of them were admin:admin. He brute forced them.

**Steve:** Yeah, that took a lot of work. So as we know, port 23 is the so-called "well-known port" for the telnet service. And as I'm sure our listeners know, telnet is the original old-school remote console access service that's been around from the start of the Internet. And in fact, look at its low port number. It's port 23; right? So you know it's been around since the beginning.

It is such a massive perennial security risk that modern systems won't even allow the telnet service to be used remotely, not even for an instant. I've been using FreeBSD Unix as my preferred non-Windows Unix platform for years. Back on FreeBSD v5, which was

the previous system I set up 15 years ago to host GRC's newsgroups, and we run our DNS server there, back then you had to really struggle to get the system to allow telnet connections. I mean, so 15 years ago FreeBSD was saying, oh, no, no, no, no. I mean, like it wasn't on by default. If you turned it on, it would fight you. If you tried to bind it to a public interface, it just really refused.

So I was curious over the holidays when I was setting up GRC's new replacement Unix machine, which is running FreeBSD v12.1. Turns out there is no way to do it any longer. The OS simply and wisely refuses. The only way to connect externally now is over port 22 using, as you mentioned before, Leo, secure shell, SSH. But not all operating systems are designed correctly. And I would argue today that is correct design. There's just no way because, I mean, it's not like there aren't secure shell clients for everything. There are. So why would you not use a secure shell connection which, as you mentioned, Leo, is encrypted, and which also has all kinds of very strong authentication measures. As I said, not all operating systems are yet designed correctly. And various IoT devices are the worst offenders.

Consequently, news was made last week when the long-time operator of a DDoS-for-hire service, a so-called "booter" service because it boots the target, the DDoS target off the Internet, published the remote login telnet - and get this Leo - username and password combinations, along with the IP address. That is, this is not just a username and password list. This is more than half a million, here is your telnet, the IP of the device, with its username and password. It turns out it was 500, little more than 515,000 devices of all kinds, which are listening on the Internet. And yeah, some of them may have IPs that float around a little bit. But more than half a million. So, yeah. The brain-dead telnet service, providing what is effectively full remote console access, is still actively supported and being served by a huge number of insecure and now suddenly much less secure Internet-connected devices.

This guy was asked why he published such a massive list of essentially what are either bots or about to be bots. And he said he had upgraded his DDoS service from working on top of IoT botnets to a new model that relies on renting high-output servers from cloud service providers. In other words, he'd apparently moved on to commandeering bigger game, so he thought he'd just drop off a little gift.

Anyway, Leo, last week you mentioned that you occasionally use GRC's ShieldsUP! service. We talked about how it had 103 million uses last time I looked. So I'll note that since telnet runs over TCP, and since ShieldsUP! checks all, well, checks for and reports on all listening ports from 1 to 1023 - no, actually it goes a few above that now. I think it goes like 15 or 18 ports above that. It crosses the 1023 barrier. I'll note that, anyway, our users can quickly use ShieldsUP! just to verify that no device on their internal network may have surreptitiously used UPnP to open a telnet port through their routers.

As we've said from the start, Universal Plug and Play is an extremely mixed blessing. In order for it to work, it needs to be UI free. And that's its whole point. But that also allows anything on your internal LAN to silently open incoming ports without permission. And I wouldn't put it past some poorly designed webcam to use your router's Universal Plug and Play to open an incoming telnet port. And who knows, it might have a non-default or a default for its make and model username and password to allow remote admin from the mothership.

Anyway, you just want to make sure nothing has opened a port into your network. And ShieldsUP! can do that very quickly. So anyway, 515,000 devices, username, password, and IP, dropped onto the Internet. So again, you don't have to be anything more than a so-called "script kiddie" in order to take that and literally write a script to see whether you're able to log onto any of those devices. And once you have, well, you've got telnet

access to something. What a world. I'm surprised that, I guess, I was going to say I'm surprised that ISPs are still allowing...

**Leo:** They should just block that port. But there are good reasons, well, there's no good reason to use it. But there's reasons they put it in, for remote access and stuff.

**Steve:** Yeah. I mean, it is a bona fide - although they are blocking Windows.

**Leo:** SFTP and port 138 or 139, yeah.

**Steve:** Yup.

**Leo:** Yeah. I guess they could reasonably block telnet port. But, yeah, I mean, who really uses that?

**Steve:** So we have a welcome micropatch for the Windows IE JScript.dll zero-day vulnerability. And I was hoping that was going to happen. We talked about this last week. This is the somewhat worrisome zero-day vulnerability. IE is able to invoke a vulnerable JScript.dll which contains a remote code execution vulnerability. We've talked about this company before, 0patch, the numeral 0patch.com. Their deal is they'll, when a Windows vulnerability is announced for which Microsoft has not yet created a patch, these guys will create what they call a "micropatch," which is typically just exactly that. It's not replacing the whole DLL.

In this case, it's 18 bytes, an 18-byte micropatch which prevents the invocation by IE of this particular DLL. I heard, well, we know both anecdotally - I ran across some reports in GRC's own newsgroup and I've seen coverage of this - that, for example, killing all access to this JScript.dll keeps Windows Media Player from being able to operate because Windows Media Player, of all things, has the option to allow web content to be displayed. And so it attempts to invoke the DLL. When it's unable to, it complains.

So anyway, their blog posting is titled: "Micropatching a Workaround for CVE-2020-0674." And they said, and I'm going to share this, the top of their blog posting - it was long - but to give our users a sense for this and because it might end up being useful in the long term for those one out of four systems that are still running Windows 7.

"Last Friday," they wrote, "Microsoft published an advisory about a remotely exploitable memory corruption vulnerability that was reported to them by Qihoo 360 as being exploited in the wild. These attacks were reportedly limited, so Microsoft decided not to rush with issuing a patch, but rather will provide one as part of February's Patch Tuesday. They did, however, provide a workaround."

They said: "Because the provided workaround has multiple negative side effects, and because it is likely that Windows 7 and Windows Server 2008 R2 users without Extended Security Updates will not get the patch at all" - since support, as we know, ended this month, they said - "we decided to provide a micropatch that simulates the workaround without its negative side effects. The vulnerability," they wrote, "is in JScript.dll, which is the scripting engine for legacy JScript code." They said: "Note that all non-legacy JScript code" - whatever that might be - "and all JavaScript code gets executed by the newer scripting engine implemented in JScript9.dll," which as we said is not vulnerable.

They wrote: "Microsoft's workaround comprises setting permissions on JScript.dll such that nobody will be able to read it. This workaround has an expected negative side effect that, if you're using a web application that employs legacy Jscript, and can as such only be used with IE, this application will no longer work in your browser." They said: "There are also several other negative side effects. Windows Media Player is reported to break on playing MP4 files. The sfc (Resource Checker tool) that scans the integrity of all protected system files and replaces incorrect versions with correct Microsoft versions, chokes on JScript.dll with altered permissions. Printing to Microsoft's 'Print to PDF' is reported to break. Proxy Automatic Configuration (PAC) scripts may not work."

So they said: "Microsoft's advisory states that the provided workaround will have to be reverted when they issue a patch for JScript.dll. However, note that some additional side effects may result from changing the ownership on JScript.dll."

Anyway, so then they go on to talk about a test case. Leo, we were talking about what it took to invoke it. And in fact they show a little bit of HTML here that I have in the show notes where it's as simple as saying <script language="Jscript.Encode">. That's all it takes to say that's the script engine we want. And in their little test, they enclose in the script tags alert("JScript.dll was loaded"). So that if you run that HTML, and JScript.dll is available, it'll just pop up a little alert dialog saying it was loaded. And so they use that in order to test their micropatch.

Anyway, so they have a micropatch. It's free. Anybody who encountered the Windows Media Player problem could revert the permissions, apply this free patch, and you'd have the problem fixed. And if you're a Windows 7 user, you could revert the fix, apply the patch, and you're good to go moving forward. So they said, and this is what I thought was interesting, they explain that they've ported this patch to Windows 7 and 10 workstation and server editions. And they're promoting their free personal and for non-profit use service.

They say: "If you're a 0patch user, you already have this micropatch downloaded to all your online computers with the 0patch Agent and, depending on your settings, already automatically applied and protected to all processes using IE11 engine for rendering content. This includes Internet Explorer, Microsoft Word, Microsoft Outlook, and a variety of other applications."

Then they said: "As with all our micropatches, you can switch this micropatch on or off and have it instantly applied to, or removed, from running applications, effectively making it a kill switch" - but it's a selective kill switch because it only does it for IE11, and that doesn't affect things like Windows Media Player - "a kill switch for JScript.dll."

And so since I think this is cool, I wanted to share a couple of the Q&A. They ask themselves: "Why would we apply your micropatch instead of Microsoft's recommended workaround?" They answer: "Our micropatch is designed to avoid negative side effects of Microsoft's workaround. It can also be easily reverted, unapplied, with a switch of a button without leaving any traces, while the workaround changes the ownership on JScript.dll."

They ask themselves: "Will Microsoft provide a patch for CVE-2020-0674 to Windows 7 and Windows Server 2008 R2 users without Extended Security Updates?" And they replied: "We don't know, but these systems are now officially out of support, and Microsoft has historically only issued security patches for unsupported systems in extreme cases." Then they quote the Shadow Brokers leak and BlueKeep. They said: "We at 0patch have committed to provide post" - and here's the point. "We at 0patch have committed to provide post-end-of-service security micropatches for Windows 7 and Windows Server 2008 R2 for three additional years, which is why we're also issuing this micropatch for these platforms."

And they said: "What will happen if I apply your micropatch, and then apply Microsoft's patch after it comes out?" They said: "When Microsoft issues a patch for the vulnerability, we'll inspect it and decide whether to replace our current micropatch," they said, "which resides in mshtml.dll and disables JScript.dll entirely, with a more targeted micropatch in JScript.dll, which will only fix that vulnerability but keep JScript.dll available." They said: "It might happen that we do so on supported Windows platforms, but keep the current micropatch on Windows 7."

Anyway, so here's the deal. These are good guys. I've never run across any downside or negative news about them. What they do is, until a patch is available, they will do a selective disabling. They've got a system on extended security updates. When Microsoft patches this, they're going to take a look at it, in the same way that bad guys reverse engineer unknown problems. If it's something they can fix, they're going to issue a true patch for Windows 7 and Windows Server 2008 R2, assuming that Microsoft opts not to break their own statement of not keeping those systems updated.

Which suggests to me that anybody who wants to stay with Windows 7 and Windows 7 2008 R2 could add this 0patch agent to their system and be kept secure moving forward. As Microsoft creates patches for the Windows 7 that is receiving Extended Security Updates, these guys are going to reverse engineer them and create a flow of monthly micropatches to keep Windows 7 security current for the same three years that paying for Extended Security Updates will do so. So we're certainly on this podcast going to keep our eye on this, and we will let our listeners know if this continues to work and how it goes. But it's a terrific free opportunity. And again, a free service for free and non-profit organizations or individual use.

**Leo:** Why are they doing this?

**Steve:** They do have an upsell. They have, if you go to 0patch.com, there is a paid service which is how they're making their money. So if an organization wants to do it, if you want to deploy it to all of your systems and so forth, then I would recommend doing it that way. But for our listeners who want to keep their home systems current, I have a feeling we're going to be recommending this moving forward.

**Leo:** I'm always nervous about free stuff. You saw what Avast has been doing with all the people using their free antivirus.

**Steve:** Uh-huh. Yes, yes, we talked about that.

**Leo:** Geez, Louise.

**Steve:** It was awful.

**Leo:** There's new stuff on it. It's much worse, yeah.

**Steve:** Oh, no kidding. Agh.

**Leo:** Yeah. It was watching everything you did, everything. And it wasn't that extension. It was just they were selling it to a - I don't know if it's a third party or a branch of Avast that collects this information.

**Steve:** Yeah, we did talk about this. They purchased an organization that was then selling this intelligence information. And, yeah.

**Leo:** Nasty, yeah.

**Steve:** And really profiting.

**Leo:** Yeah.

**Steve:** So Leo, a little bit of miscellany. I forgot to tell you last week that we made another attempt at Star Wars. You had said, and you were right...

**Leo:** Oh, yeah, because you went to that crappy...

**Steve:** Oh, my god, that 4DX.

**Leo:** ...amusement park theater version.

**Steve:** The vomitorium Star Wars. Oh, my lord.

**Leo:** So absent the seats moving and the spray in your face, what did you think of the movie?

**Steve:** I loved it.

**Leo:** Yeah.

**Steve:** Absolutely loved it.

**Leo:** Yeah. I mean, I understand the criticisms. But we're completing a nine-movie cycle here; right?

**Steve:** Yeah. I mean, we were kids when the first one came out.

**Leo:** Literally, yeah, '77, yeah.

**Steve:** So, yeah. I did like it. And I did resign my Disney Plus subscription before - for me it was on the 24th it was coming up, and that's why Lorrie and I watched that two-hour making of the first Star Wars trilogy that was really good. We're really glad we watched that.

**Leo:** Good.

**Steve:** And now I have to mention "Picard."

**Leo:** Uh-oh.

**Steve:** Which was, I thought, wonderful also.

**Leo:** Oh, good. Oh, good.

**Steve:** It's 8.8 on IMDB, which is not easy to get. And it's released, an episode drops every Thursday on CBS All Access. Non-U.S. Prime users can get it a day later on Amazon Prime. So if you're using a VPN to appear to be somewhere else, then maybe you can get it through Amazon Prime. I have CBS All Access. It's one of my cord-cutter subscriptions. It really looks good. We're going to watch this first episode a second time just because this guy can really act. And lots of special effects, lots of action. Looks like it's going to be an interesting series. So I wanted to tell our listeners that I give it a thumbs-up, even though it's not as easy to do it because you've got to buy Access in order - unless you have some other way of finding it.

And at grc.sc/, you know, "sc" as in shortcut, grc.sc/roadmap is a recently created roadmap for the future of SpinRite. So for anyone who is curious, grc.sc/roadmap. I lay out where we are, what I'm going to do with 6.1. And a new wrinkle was added because I'm in the process of firing this up. It turns out all of my SpinRite tools from, what was it, 2013 were running under Windows XP, and they were 16-bit. So I'm having to change a bunch of things around.

It turns out that finding a linker that will run under Windows is not easy. But Watcom looks like the right solution. The Watcom system is a beautiful piece of work that doesn't really get much attention. It's known within the development community as a fabulous compiler, and it's got really broad support. So I'll probably be using it for remote debugging, which I wasn't doing before. I was using a native debugger, a 16-bit debugger. Well, naturally, because it's on DOS, which is where SpinRite runs.

But anyway, the point is that somebody referred to AHCI, the Advanced Host Controller Interface, which is what SpinRite 6.1 will be supporting natively, as "legacy." And I thought, whoa, wait a minute. When did that happen? Well, it turns out that it's legacy inasmuch as it is fixed, it is set in concrete and is not changing any longer. But of course the newer controller is the NVME controller, the Non-Volatile Memory controller, which non-volatile memory built into the latest laptops and like my little Intel NUCs that I've been talking about, they're all NVME. So of course that's going to need native support, too.

So anyway, I realized, okay, I need to sort of lay out the sequence of work that I'll be doing and how I see the future. So anyway, I created a roadmap for my ongoing development work. I have a feeling, I mean, I'm, like, I'm back in. I can't say I've

actually started, but within a couple days. The problem is I've had to, like, really, my whole development environment has had to be reengineered in this era of 64-bit OS which is trying to reach down to a 16-bit real mode DOS and do development work there. But anyway, I've figured out how to do it. I just haven't finished putting all the pieces together yet.

**Leo:** Cool.

**Steve:** For those who are interested, we are getting there. SpinRite 6.1 is on its way.

Okay. SHAmbles. What happened? There's been a duo, Gatan Leurent and Thomas Peyrin. We've spoken of them before because for years they have been methodically pounding on and essentially chipping away at the aging SHA-1 hash. And in the course of all of that, they've come to know it extremely well. And in a paper just published, they have finally broken SHA-1. They've taken it from weakened to, well, I would say "weakened and worrisome" to "demonstrably broken." Their recently released paper is titled "SHA-1 Is a Shambles. First Chosen-Prefix Collision on SHA-1 and Application to the PGP Web of Trust." And I'm going to share with our...

**Leo:** Oh, interesting.

**Steve:** Yeah, yeah, yeah. They created a practical attack on PGP. Essentially broke its, well, to the degree that PGP still supports SHA-1. So not SHA-256 signatures, but SHA-1. So to kind of get a look inside, I'm going to share the abstract from their paper because it's just got a whole bunch of cool bits. They said: "The SHA-1 hash function was designed in 1995 and has been widely used during two decades." They said: "A theoretical collision attack was first proposed in '04, but due to its high complexity it was only implemented in practice in 2017." So isn't that cool? First of all, a theoretical collision attack proposed in 2004. But due to its high complexity - that is, computational cost - was only implemented in practice, that is, we only got the computation resources 13 years later, in 2017, using a large GPU cluster.

They said: "More recently, an almost practical chosen-prefix collision attack against SHA-1 has been proposed. This more powerful attack allows to build colliding messages with two arbitrary prefixes, which is much more threatening to real protocols." And I'll explain why in a second. They said: "In this paper, we report the first practical implementation of that attack and its impact on real-world security with a PGP/GnuPG impersonation attack. We managed to significantly reduce the complexity of collisions attacking against SHA-1. On an Nvidia GTX 970, identical-prefix collisions can now be computed with a complexity of" - and I'll explain what these things mean in a second - "the complexity of 2^61.2 rather than 2^64.7."

Okay, now, think about that. So 2^62.2 versus 2^64.7. So that's a difference of a little more than three, which is to say 2^3 is 8. So it's a factor of 8 reduction in complexity. And chosen-prefix collisions with a complexity of 2^63.4 versus 2^67.1. Okay. So there, 63 versus 67, that's a difference of 4, so 2 to that is 16. So they made it 16 times more practical.

So they said: "When renting cheap GPUs, this translates to a cost of $11,000 U.S. in the first case, for a collision, and $45,000 for a chosen-prefix collision within the means of academic researchers." Okay, but also notice that what they did was, by getting that factor of 16 reduction, they were able to - I didn't do the math beforehand, but I've got a calculator right here; 45K was $720,000 for one collision. So, yeah.

**Leo:** It's not easy. Not cheap.

**Steve:** Yeah. $720,000 of computing.

**Leo:** You've got to really want to get in.

**Steve:** For just one collision.

**Leo:** That's a NOBUS, yeah.

**Steve:** Yeah. So they dropped it from 720,000 to 45,000. Still not cheap, but within reach.

**Leo:** It shows you, I mean, that's a good measure is the cost of computing power, yeah.

**Steve:** Yes, yes. So they said: "Our actual attack required two months of computations using 900 Nvidia GTX 1060 GPUs." So again, consider how high the threshold is. I mean, when we say SHA-1 has been compromised, that is, it's broken, well, it's like it's now to the point where we can no longer trust it. But that's how high the trust bar is, that to create an academic collision with a whole bunch of special case caveats, in this case this chosen prefix collision, it took 900 Nvidia GTX 1060 GPUs. Oh, and they said: "We paid $75,000 because GPU prices were higher at the time. And," they said, "we wasted some time preparing the attack."

So they said: "Therefore, the same attacks that have been practical on MD5" - remember that's the much smaller hash, like way older hash, MD5 - "since 2009 are now practical on SHA-1. In particular, chosen-prefix collisions can break signature schemes and handshake security in secure channel protocols such as TLS and SSH. We strongly advise to remove SHA-1 from those types of applications as soon as possible." That is to say, signatures and secure handshakes.

They said: "We exemplify our cryptanalysis by creating a pair of PGP/GnuPG keys with different identities, but the same SHA-1 certificates. An SHA-1 certificate of the first key can therefore be transferred to the second key, leading to forgery. This proves that SHA-1 signatures now offer virtually no security in practice. The legacy branch of GPG still uses SHA-1 by default for identity certifications. But after notifying the authors, the modern branch now rejects SHA-1 signatures." And they said the issue was tracked as CVE-2019-14855.

So then I'll just finish with a little bit of background, which is fun, from their paper, since we have got time. They said in their introduction: "Cryptographic hash functions are present in countless security applications and protocols, used for various purposes such as building digital signature schemes, messaging authentication codes, or password hashing functions. In the key application of digital signatures, for example, hash functions are classically applied on the message before signing it, in order to improve efficiency and provide security guarantees. Informally, a cryptographic hash function H is a function that maps an arbitrarily long message M to a fixed-length hash value."

And they say: "We denote n its bit size. Collision resistance is the main security property expected from a hash function. It should be hard for an adversary to compute a collision." In other words, two distinct messages, M and M0, that map to the same hash value H(M) and H(M0), where by "hard," they have in quotes, one means no faster than the generic 2^n/2 computations birthday attack.

So in the case of SHA-1, it's a 160-bit hash. So 160/2 is 80. So it should not be possible to do this any faster than 2^80. And as we see, they've brought it down through extensive analysis down to the region of 2 to the, what, 60s, 62, 63. So what is that, a 2^14 reduction. Well, 2^14 is a big number. 1024 is 10, so 16. So it's 16,000. So they've reduced the complexity by an order of - on the order of 16,000 making - and so we saw even now it's still $45,000 in GPU cost. So multiply 45,000 times 16,000, and we're into some serious money. Which means, you know, as long as there was no weakening of the hash, it was sufficiently strong. Unfortunately, it hasn't stood the test of time.

Anyway, they said: "A cryptanalyst will try to find a collision for the hash function at a reduced cost, but ad hoc collision attacks are hard to exploit in practice because the attacker has then usually little control over the value of the actual colliding messages, in particular where the differences are inserted, which are the interesting parts when attacking a digital signature scheme, for example."

They said: "Thus, one can consider stronger and more relevant variants of the collision attack in practice, such as the so-called chosen-prefix collision or CP collision." Okay, which is two message prefixes, P and P0, are first given as challenge to the adversary, and his goal is to compute two messages M and M0 such that the hash of P concatenated to M equals the hash of P0 concatenated to M0.

"With such ability, the attacker can obtain a collision even though prefixes can be chosen arbitrarily, and thus potentially contain some meaningful information. A chosen-prefix collision can also be found generically," as they note, "with 2^n/2 computations. And there we're back to 2^80," they said, "for a 160-bit hash function like SHA-1. But ad hoc chosen-prefix collision attacks are much more difficult to find than plain collision attacks because of the random and completely uncontrolled internal differences created by the prefixes. Yet a chosen-prefix collision attack was found for the MD5 hash function, eventually leading to the creation of colliding X.509 certificates, and later a rogue Certificate Authority. CP collisions have also been shown to break important Internet protocols, including TLS, IKE, and SSH, because they allow forgeries of the handshake messages."

So anyway, I won't go on at that level. Their paper is online. I've got a link in the show notes if anyone's interested. Essentially what this means is that, through a lot of work, I mean, serious academic research, these guys have conclusively demonstrated that, even though it's still expensive, it is no longer credible to trust anything which is protected by SHA-1. Although there are still limitations. It's still expensive. It's not like it just, like, falls out instantly. No cryptographic protocol or cryptographer worth their salt would recommend using SHA-1 now. It is the case that you cannot trust it.

**Leo:** Even if it's salted?

**Steve:** Even if it's had salt sprinkled on its tail, yup.

**Leo:** Aha.

**Steve:** And the good news is the cryptographic industry, as we all know, anticipated this.

**Leo:** Yeah.

**Steve:** We've had SHA-2, which comes in various flavors, SHA-256, what is it, -384 and -512, they all exist. They're there. They're in place. They're ready. And we already have SHA-3 waiting in the wings, although no one thinks we need it yet because SHA-256 is very strong.

**Leo:** So all of these, it's really just inevitable, as long as processing power increases exponentially, that they're going to get cracked eventually.

**Steve:** Yup. I mean, they're doing something that is "hard," in quotes, but our definition of what is "hard" keeps changing.

**Leo:** Yeah, it gets easier.

**Steve:** Now we're got facial recognition happening on the street corner of everyone who walks by. That was inconceivable 20 years ago.

**Leo:** Yeah.

**Steve:** But now it's like, oh, yeah.

**Leo:** It's kind of amazing, yeah. Ah. Once again, another gripping, thrilling edition of Security Now!, Steverino. Thank you.

**Steve:** Well, the spouses of some of our listeners may disagree.

**Leo:** Oh, they're all asleep by now. Hey, I do want to remind people we do want you to participate in our survey.

**Steve:** The survey, yes.

**Leo:** Want to make sure the Security Now! listeners are well represented. The audience survey takes a few minutes. We don't ask for email. Because we don't track you, we don't have any way to know anything about you. So by doing this survey once a year, we get a better idea. It helps us prepare programming that you're going to like, but it also helps us sell ads, and that's our lifeblood. So twit.to/survey20, if you don't mind. You don't have to answer any question you don't want to answer. You don't have to answer the survey at all. But it's nice if we can get some of you to do that, anyway: twit.to/survey20. Thank you, as always, for supporting Security Now!.

Steve's site is GRC.com, the Gibson Research Corporation. That's where you'll find SpinRite, the world's best hard drive maintenance and recovery tool. Everyone needs a copy of SpinRite. And while you're there, of course, you can pick up a copy of the show, 16 and 64Kb audio versions there. A great transcript. Takes about three or four days for Elaine to put that up after the show.

**Steve:** She gets the audio, typically I compress it early in the morning on Wednesday. I get the transcript from her early, like mid-morning on Thursday. And then, if I'm not distracted, I get it up immediately; or it's delayed depending upon when I see her email and blah blah blah.

**Leo:** I don't want to know how you get distracted.

**Steve:** Yeah.

**Leo:** Oh, look. It's a Fibonacci series. GRC.com, that's the place to get all of that. You can also find many other free and useful tools like ShieldsUP! and everything. So check it out, GRC.com. Steve's on Twitter at @SGgrc. DM him there if you have questions, comments, suggestions. His DMs are open.

You can also get the show from our website, TWiT.tv/sn. Audio and video available there of every show, all 751 of them now. Or on YouTube. You can watch it on YouTube, of course. Best thing to do, if you ask me, subscribe. Find your favorite podcast application and subscribe. Then you don't even have to think about it. It's just on your device, ready to listen to whenever you're in the mood. And who isn't always in the mood for Security Now!? I've always got time for Security Now!. Thank you, Steve. I'll see you next week on Security Now!.

**Steve:** Just like Jell-O. There's always room.