DEF CON & Black Hat, Part 1

Description: This week, following the DEF CON and Black Hat conferences, Leo and I catch up with the past week's crazy news, including a distressing quantity of distressing Win10 news, Apple's changing bug bounty policy, newly disclosed Android takeover flaws, yet another way to track web visitors, hackers spoofing Tesla auto sensors, Firefox and LastPass news, and some miscellany. Then a 19-year-old stubborn decision by Microsoft comes home to roost, and a handful of new problems are found with HTTP.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-572.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-572-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. He's been reading all the releases from DEF CON and Black Hat, all the new security exploits. We'll cover those and a whole lot more. This is going to be a jam-packed great episode. You stick around. Security Now! is next.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 572, recorded Tuesday, August 9th, 2016: DEF CON and Black Hat, Part 1.

It's time for Security Now!, the show where we talk about your privacy and security online with this fellow right here. His name happens to be the illustrious Steve Gibson. Hi, Steve.

Steve Gibson: Leo, this is...

Leo: Peace in our times.

Steve: ...the penultimate episode of Year 11.

Leo: You used the word correctly in a sentence. You have grasped the meaning.

Steve: I forced myself to because I have clearly put on record it's a useless word that no one has a need for.

Leo: You know what else? This is also the penultimate show in this old studio. Next week will be our last show in the Brick House. And the week following we'll be in the Eastside Studios.

Steve: And from what you've said, our listeners are not going to notice any big difference, like looking at you there, where it's like, oh.

Leo: It should look the same. The desk will be cleaner. But everything you see here - and most people, by the way, it's a tiny percentage that actually watch the show. But most people who watch the show, everything you see here you will see in the new - it'll look the same. There may be some subtle differences, but it will look the same. Actually, one of the differences is right now I'm in a fishbowl. We used to have curtains all the way around and so that nobody could see. But they took the curtains down, and now I'm in a fishbowl.

Steve: And we've commented about how much brighter the sound was.

Leo: Yeah, looks different, yeah. The new studio will be...

Steve: No, I mean...

Leo: It sounds different, yeah. But the new studio will be drywall around three walls of it, but there will be windows to the outside world in my new studio, so that's cool. Unlike this studio. So anyway, it's just different. But we're going to take all these pieces with us. Which is, I think, okay, because I don't think the people who are building a brewery in here want them.

Steve: Oh, so you're going to actually move your office installation.

Leo: Yeah.

Steve: Into the other building.

Leo: Yeah, exactly.

Steve: Nice.

Leo: All the stuff that's here, going.

Steve: Nice. Except apparently some of the books you're leaving behind.

Leo: Well, I don't have room for as many bookshelves, so I have to go through them. But, you know, programming books from the year 2008 aren't that useful.

Steve: You know? And...

Leo: You have worse.

Steve: I have, you know...

Leo: You have a worse situation.

Steve: ...DOS internals books. And actually I did, I did need them when I did SpinRite 6 because I was doing new things with the VGA adapter that I had not done before. And you can't find anything on programming the VGA adapter any longer. So, I mean, even online it was scarce. So I was glad that I have sort of an archeological environment here. So for me, yes, a little bit different.

Leo: And you're never moving. How long have you lived in that apartment?

Steve: Since, well, this condo since 1984.

Leo: Okay. So, yeah. And there's no reason to think that you will be at any time required to pack those books up again.

Steve: No. That'll be somebody else's problem.

Leo: So if I were in that situation, I would not ever get rid of a book. I love books. I hate to get rid of books.

Steve: I do, too. Although, Leo, they're just not practical anymore.

Leo: Not at all.

Steve: I mean, like, when was the last time I actually went to a book, rather than just typing a phrase into Google? And there's, like, everything I could possibly want to know, and much more current, and even some commentary. I mean, just it's all changed.

Leo: I can only imagine, though, the look on the guy's face who wanders over to our used bookstore and finds the best collection of old Python books ever.

Steve: Well, for the right person...

Leo: Web 2.0. For the right person, it's a treasure trove. And these were expensive books. There's probably $10,000 worth of books behind me.

Steve: Remember how expensive technical resources used to be?

Leo: Yeah. They really were expensive. Now it's all online.

Steve: I'm seeing that actually a lot in medical stuff that I'm having to buy sometimes. I have some $300 textbooks that I've purchased. And it's like, ooh, do I really need this? It's like, well, apparently I do.

Leo: Or how do they get away with this? You know?

Steve: Yeah. So we are post-DEF CON and Black Hat, with an overflowing cornucopia of new horrors to share. So many, in fact, that I called this Part 1. Even as I was sort of putting this one to bed, I looked up in my Twitter feed, and I found a whole bunch of other things that it's like, oh, no, okay. We're not done yet. So all kinds of fun things. I do have - I do not want this to be the "pound on Microsoft for Windows 10" podcast. That's not what this is going to become. I've said my piece about Windows 10. I've implemented my piece with Never10. By the way, it just crossed 1.8 million downloads. Naturally, it's slacked off a lot since they stopped pushing Windows 10 on everybody, but still 5,000 people a day for some reason think they need it.

Leo: Why are they downloading it now?

Steve: I have no idea. No idea. Although we still get downloads of the Click of Death podcast and things. So it's like, I think, for some people, they're like you and I with books. They're like, oh, I might need this someday.

So we do have a distressing quantity of distressing Windows 10 news. I heard you mention on MacBreak Weekly, and we'll cover it here, Apple's changing bug bounty policy, which is nice. Some newly disclosed Android takeover flaws, some that have been patched. Some won't get around to being patched till next month because they happened more recently. Yet another clever and sad and distressing way to track web visitors around the Internet. At DEF CON, some hackers spoofed Tesla's automobile sensors, which I kind of think is a nonstory, but I want to explain why.

Leo: God, I hope so.

Steve: We have some Firefox news, some LastPass news, some miscellany. Then three more things I stuck afterwards: a 19-year-old stubborn decision by Microsoft which has come home to roost, and then two different groups have found new problems with HTTP and HTTP/2. And most of this is courtesy, but not all, courtesy of DEF CON and Black

Hat. And as I said, I was looking up at my feed, and there's, like, even some more interesting-looking things. So, and one of these, in the case of this 19-year-old stubborn decision by Microsoft, this one I'm going to have to take a look at. I think we're going to have some horrified listeners.

Leo: Uh-oh.

Steve: Not that we don't usually. But extra horrified. So a great podcast, our penultimate Year 11 podcast.

Leo: So when you say "penultimate Year 11," you mean we'll be beginning Year 12 of the show in two...

Steve: In two weeks.

Leo: Wow.

Steve: Yes.

Leo: Wow. And it'll be easy to keep track of, now, how long we've been in the new studio, because it's coincident with the move to the new studio. That's good. We begin our 12th year in our third studio. Wow.

Steve: And each one's lasting much longer than the one before.

Leo: This one should last me, like you, for the rest of my life. I am not moving ever again. We signed a 10-year lease with a five-year option.

Steve: Nice.

Leo: So that, let's see, that'll take me to 74. And maybe I'll retire in the next 15 years.

Steve: You and I will be like Jerry Pournelle.

Leo: I know. But you know what, Jerry still has a lot of things, good things to say, so...

Steve: [indiscernible] Windows 10 [indiscernible] Windows 10 [indiscernible].

Leo: [Indiscernible]. That will be fun.

Steve: So Woody Leonard...

Leo: Oh, yeah.

Steve: ...who writes the Woody on Windows column for InfoWorld...

Leo: Talk about old-timers.

Steve: Yeah.

Leo: He's been doing that forever.

Steve: Yup. His title of his most recent posting was "Windows 10 Anniversary Update Woes Continue." And I just thought I'd run through these quickly for listeners who at least will have a feeling, if they've had some of these problems, to know that they're not alone. His little posting starts out, before I paraphrase, he said: "Problems with last week's Anniversary Update keep piling up, and solutions remain elusive." And then he says: "Late last week I recommended that you actively block the Windows 10 Anniversary Update. The past few days have brought yet another wave of complaints. Here's a sample," he writes.

So apparently Windows 10 is freezing on a number of systems. And I actually think that later on I'll get to at least some of the causes of these things because, in digging into other stories, I realized, ooh, this is what's causing that problem and so forth. So there was, on Reddit, Woody was following some dialogue that had 680 comments about post-Anniversary Update freezing of all the systems. And apparently some of this involves some incompatible AV software that Microsoft, as it's explained actually by McAfee - not himself, McAfee now owned by Intel, the corporation - that Microsoft did not have time to incorporate the checks for compatibility of specific McAfee AV versions. And so it'll update over and be incompatible and thus cause a lot of problems.

But in any event, Woody notes that nobody yet has found a complete solution for these. And he notes from his own experience that Edge, the new browser, which is in many ways a good thing, we've talked about how Microsoft did, they bit the bullet and just said, okay, we cannot keep pushing IE's code base forward any longer. We just need to start again. And when you do that, you have the advantage of all the experience that you've gained. And so instead of cobbling things together, or in many cases not being able to do what you would like to do from an old code base, you can make it the way you want it. And we know that Edge has a very good, strong security model and is very fast and has a state-of-the-art scripting interpreting system. So in many ways it's good.

However, Woody notes that Edge still has plenty of problems. He says: "I've hit situations where Edge will not close by clicking on the red X. Also, I can X out of the last open tab and Edge keeps running, when closing the only open tab should shut down the program as a whole. The problems seem to appear after visiting sites with lots of ads,

like the ones," he says, "linked to from MSN.com, for example. Once the problems start, they don't go away." The only solution, he writes, that he's found is to reboot.

And then two different antivirus companies have reported problems. As I mentioned before, McAfee warns, actually with emphasis in their note: "DO NOT upgrade to the Windows 10 Anniversary Update without first verifying whether your McAfee product is compatible. This caution affects the products listed in the section above," of their notice. Then they said: "Microsoft intended to implement an upgrade and installation check to ensure that no incompatible McAfee product versions could be installed or were present. Due to time constraints, Microsoft could not implement the intended version check in the Windows 10 Anniversary Update."

So, unfortunately, that leaves people with this update, the big Anniversary Update, and this apparently causes real problems because, as we know, we've been talking about this recently, one of the things that AV tools have started doing is digging themselves very deep down into the kernel. This has been causing problems, as we know, because it can increase the attack surface of the machine. If you have a third party's code inspecting everything that comes through the network, well, it has to be flawlessly implemented, which is something that Microsoft has carefully been doing over a long period of time. If there are things like buffer overruns in that add-on code, that creates vulnerabilities. And we've covered those in the not-too-distant past on the podcast.

But as a consequence, it means that, if there are some changes that Microsoft made in the kernel, where they normally would have the right to make such changes, the idea being that it's the API layer, this Application Programming Interface, that's supposed to be the boundary, the formal boundary between applications and the OS. And the idea being that, as long as the API stays the same, the OS vendor is free to do anything it wants to behind the API because it's that layer, that way that the applications have of talking to the operating system, that needs to remain uniform. But then who cares how the job gets done?

Well, what's happened is, by essentially putting kernel drivers in the OS - and in many cases Microsoft doesn't sort of officially support the kind of things that these vendors want to do. So then they have to even break the kernel API. There is another API, for example, for device drivers, which is completely different from the application programming interface. And that API, as any, assumes certain things that driver developers would want to do and makes those OS services available.

Well, the problem is, if these vendors need to do something outside the API, they will have to hook functions in the kernel, which then makes the whole system far less stable. So reading between the lines, our advice for a while has been, unless you really have to have these things down, added to the operating system, unless there's some overriding reason, it's really becoming better not to use these. And as we've also talked about, many of these are now also putting certificates in your system and then not managing them as responsibly as we would like, which then allows for third parties to come along and create spoofed website attacks.

And then, finally, he says: "I'm still unclear" - that is, Woody says. "I'm still unclear about the ability to block [what he calls] crapware tiles." He says: "I wrote about the problem a couple of weeks ago." And actually I do know what causes this now, and we'll get to that in a second. "Admins cannot keep Microsoft from pushing crapware Live tiles onto Win10 Pro PCs because certain Group Policies don't work in the Anniversary Update."

And believe it or not, and I'm just - I'm stunned by this news - Microsoft has removed a

bunch of very useful mitigations against many of the things that people found objectionable about Windows 10. They took them out of the Anniversary Update for the Pro version, not the Educational version and not the Enterprise version. I think Enterprise they left them in because Enterprise wants control. Education they left them in because they didn't want to force educational usage to have that if they didn't want it. And that left, like, all of the rest of us, non-Enterprise and non-Educational, that is, the Windows 10 Professional people, without the same level of control over this that the other, like the very high-end to very low-end both have - which, if I were using Windows 10, would annoy me a lot.

And he writes: "My current Win10 Pro AU [Anniversary Update] machine has tiles for Solitaire, Candy Crush Soda Saga" - this is just so sad - "Pandora, Asphalt 8, Age of Empires Castle Siege, FarmVille 2, Minecraft, Twitter, and Get Office - in other words, about half of my Start Menu tiles are unabashed, Microsoft-installed crapware, all on a machine that's been through the official 'start fresh' regimen." He doesn't want those, and he can no longer turn them off.

Leo: Not true. You just right-click and you remove them. They're stubs for installing. They are not the app. That's not true.

Steve: Okay. Well, I have the registry edits...

Leo: I don't have them. Here, you want to see? You don't have to, please don't edit the registry. You right-click them and delete them. Oh, my god. You want to see my startup after Windows 10 Anniversary? None of that. None of that stuff.

Steve: Okay.

Leo: You just right-click, and you say "remove." And they're not the apps, by the way. They are ads, I admit. They're stubs that you click them, it'll take you to the store, and you download it. Also I should point out that you're not allowed to, if you took the free upgrade, deny updates. That's part, that was part of the deal. You've agreed to all updates. You can delay them. But ultimately you'll have to take the Anniversary Update. You can't not do it.

Steve: So others are reporting that the Anniversary Update is not respecting unknown partition types. They're finding that installing the Anniversary Update, for example, in dual boot environments, where they have Linux and Windows, that they lose access to Linux. And so it's common practice to install Windows first and Linux afterwards because Microsoft does have a habit of overriding the boot sector, which would cause you to lose your multiboot behavior. But this has gone further. And there's a lot of report of this on the 'Net, that Microsoft is just blowing away non-Windows partitions as part of the Anniversary Update. And the term I'm seeing is "borking" dual-boot partitions. So be advised that that could happen. You might want to make an image of the whole drive before you move, if you haven't already.

Okay. So the reason that what Woody said made sense to me was that there is extensive and pretty clear coverage of the Group Policies and registry entries which Microsoft is documenting they have changed. And they're saying, when you go to Group Policy Edit

and look at these keys, they're specifically saying that these are no longer available for Windows 10 Pro, one being turning off the Microsoft consumer experience, which controls, among other things, installation of third-party apps and extra links on Windows 10. Now, so maybe, Leo, what you're saying is that Windows will, or Microsoft will push these, and then the user is free to delete them.

Leo: That's right.

Steve: And so that means...

Leo: It's part of the default install. And so there's two things you can do. One, there's a setting that turns off advertising, if you call it that, in the Start Menu. I of course immediately did that. And then you can right-click and unpin it. And if you want to, you can uninstall it. I just did, in fact, I just checked because I had - it was hard to find something, but I found a solitaire entry. It wasn't on my tiles, but it was in the menu still, and I uninstalled it. And it's just like that, and it's gone. So, you know, Woody may not be really a fan of Windows 10. There is some disinformation that gets spread. I don't think Windows 10, I don't think Microsoft is as bad as, I mean, look, I only have one machine left with Windows 10 on it, so I can answer questions like this. I far prefer Linux. But it's not that bad. You can remove those.

Steve: Okay. So for what it's worth, for people who are using Windows 10 and are technically sophisticated, I have here in the show notes a number of, I mean, a detailed itemization of the Group Policies which have been removed from Windows 10 Pro. They still exist in the educational version, and they exist in the Enterprise version. They are gone from Windows 10 Pro. They used to be there. They've just been taken out.

So you're no longer allowed to turn off the Microsoft consumer experience. You cannot turn off showing Windows tips. You have less control over the lock screen. And you're no longer able to prevent changing the log screen and the logon image, nor can you disable all apps from the Windows Store, which are features that the other versions of Windows don't support. So if anyone is curious and wants more details, I've got it all laid out here in the show notes.

And then Mary Jo finally, you know, we talked last week about the whole assistive upgrade backdoor. Mary Jo reported that - and so I wanted to share this information with our listeners who might find it useful, that Windows 7 and 8 unused product keys can still be used to install Windows 10.

Leo: Yeah. I saw that. That's cool.

Steve: Yeah. So she asked Microsoft, like, okay, do you guys know about this? And they said, uh, no comment. And she said, well, okay, is it going to go away? Uh, no comment. And so for what it's worth. She said, and this is last week, she said: "In spite of the official end of the free Windows 10 update offer on July 29, it seems that any valid Windows 7/8.x retail product key still installs Windows 10 for now." And again, this, obviously, this is sort of off-the-books behavior, not what anyone is expecting. And so no idea how long that will last. And so she has, you know, she's verified with Microsoft that this is happening, and users are able to install Windows 10 using Windows 7 and 8. And I

said "unused product keys." But it looks like even valid product keys. So anyway, so that, again, also.

Apple has changed their - and they announced this just last week, on Thursday, at Black Hat. This is Ivan Krstic. Is that how you pronounce his name, Leo, do you know? K-R-S-T-I-C? Krstic?

Leo: It's either Krstic or Krstic. I don't know.

Steve: Okay. He announced a reversal of Apple's longstanding "we don't pay bug bounties" policy, with the news that Apple will begin offering cash bounties of up to $200,000 to researchers who discover vulnerabilities in its products. And I know you guys just talked about this on MacBreak.

Leo: I like the rationalization. Keep going.

Steve: Yeah. Well, and to me it makes sense because - so what Apple was saying previously was that they weren't going to be in a bidding war with government institutions, for example. I mean, for example, we know that the FBI reportedly, or it's a rumor, but it's generally agreed that they paid something near a million dollars for that hack which allowed them to get into Syed Farook's work-related iPhone after the San Bernardino shootings.

So a CEO of Securosis, Rich Mogull, he said: "A bug bounty program is unlikely to tempt any hackers who are only interested in getting a massive payout. For those who only care about cash, Apple could probably never pay enough," meaning they wouldn't be able to outbid somebody who had a nefarious application for a breach. "But for those who care about making an impact, getting a check from Apple could make all the difference by incentivizing good work." And to me that makes sense.

And so I paraphrased it. I said: "Put another way, if you have no interest in allowing your work to be used for evil, but you would like your important security findings to be rewarded and supported, that can now happen on Apple platforms." So from my standpoint I think it makes absolute sense for Apple to say we'll pay a reasonable amount of money because - and as I understand it, part of this is a whole change in their approach to security researchers; right, Leo? They're, like, saying we're going to - we'd like to work with people to find problems.

Leo: Good idea.

Steve: What a concept.

Leo: I like the -I don't know if this is in the release you're reading, and I didn't read this, Rene reported it. They said, well, it's getting so hard to find problems in Apple software now that we want to reward you for doing that.

Steve: Yeah, [crosstalk].

Leo: They think it was easy before, so we didn't want to reward you? I'm not sure that they're saying.

Steve: Well, and we keep seeing...

Leo: There's plenty.

Steve: ...rootkit hacks. I mean, no matter what Apple does. In fact, we just had that 9.3.3 went to 9.3.4 to remove another way of getting into the phone. So it's like, good luck. I mean, again, we know these things are just too complicated. The harder you press, the more problems you can find. I think this is great.

Leo: Yes.

Steve: I'm glad that Apple is saying, yeah. White hat hackers, I mean, this stuff does take time. I look at this kind of stuff, and I think, wow, that would be fun to do. But I don't have a couple months to, like, just find a big problem and then say, here, Apple, and have them fix it. It's like, okay. If it were a profit center, and I didn't have a lot of other things to do, then I think that would be a really fun way to hack. So I think it's great.

Leo: Yeah. And after all, don't you want, I mean, as users, don't we want Apple to incent people to try to find bugs?

Steve: Yeah. I mean, the whole Pwn2Own competition [crosstalk] happens, which is finding all these problems. And they're all responsibly disclosed, and they're fixed by the time we find them. But until we found them, they were potential zero days that nobody knew about.

So there are - an Adam Donenfeld of Check Point presented at DEF CON. His presentation was titled "Stumping the Mobile Chipset." And the little synopsis of his presentation read: "Following recent security issues discovered in Android, Google made a number of changes to tighten security across its fragmented landscape. However, Google is not alone in the struggle to keep Android safe. Qualcomm, a supplier of 80% of the chipsets in the Android ecosystem, has almost as much effect on Android's security as Google.

"With this in mind, we decided to closely examine Qualcomm's code in Android devices. During our research, we found multiple privilege escalation vulnerabilities in multiple subsystems introduced by Qualcomm to all its Android devices in multiple" - and this is a little redundant - "multiple different subsystems. In this presentation," they write, "we will review not only the privilege escalation vulnerabilities we found, but also demonstrate and present a detailed exploitation, overcoming all the existing mitigations in Android's Linux kernel to run kernel-code, elevating privileges and thus gaining root privileges and completely bypassing SELinux," which of course is the Security Enhanced Linux.

So they gave that presentation. All versions of Android were vulnerable to these newly revealed flaws. However, they had been trickling the news out responsibly since April and from April through last month. And so most of them have been fixed. In the supply chain, Google fixed all but one, which was unable to make it into the August updates. So it'll be in the September updates. So, but these flaws affect Android phones and tablets that ship with Qualcomm chips, which could let a hacker take full control of an affected device.

And I had a list of the phones somewhere. Oh, there. Google's Nexus 5X, Nexus 6, 6P; HTC's One M9 and HTC 10; and Samsung's Galaxy S7 and just the S7. Oh, and I love this, too. And the recently announced BlackBerry DTEK50, which of course BlackBerry touts as the most secure Android smartphone. And as we've often said, that's just marketing speak. No one can declare something the most secure anything. They can declare their intent. But as we have learned, because security are mistakes, you don't deliberately make mistakes. That's why it's a mistake. And those mistakes create vulnerabilities that can then be exploited. Which sort of tautologically says you can't say it's the most secure phone because it's not a statement that contains any sense.

So anyway, so these are malicious software install exploits, meaning that somebody, a malicious actor would need to sneak an app either past Google's scrutiny, or the user would have to be incented. And unfortunately we've just seen this, we were talking about this with Pokemon Go, that people were installing it sideways into their phone by deliberately turning off the only install apps from the Google Play Store, turning that off in order to install something that wasn't available. So we know that that's being done.

But anyway, so this is not something like Stagefright, where just someone sending you an MMS can take over your phone. This requires you to install some software. However, that software, that Android application needs no special privileges at all. It doesn't need to ask for anything. It can look completely benign. Yet using these exploits it's able to essentially get root on your phone and then have the run of the kingdom, do anything it wants to. A lot of them are in the process of being fixed and will be fixed, the final one from Google in the next month's batch of patches.

Okay. We've talked often about just the tracking technology. Whether you're concerned about tracking and being tracked or not, it is in my mind separable from just the technology, which is often fascinating. And I do find myself jarred when I go to a site whose ads are shockingly relevant. I mean, there's a substance that I'm exploring for the Healthy Sleep Formula known as oleamide, and it's looking very good. I haven't said anything about it or written anything about it yet, but it looks like it's another major step forward. It already exists in us. And so I have been purchasing some in order to experiment with it, as have a bunch of other people that are in a small group. And I'm finding the ads for it like in random places that I visit. And it's like, okay. That's just too weird.

I mean, so I'm sure everyone has this experience, unless you are really crazy about blocking cookies and private browsing and flushing everything. Actually, this exploit still affects you even in private browsing and can be used to track you through a VPN and across a private browsing session. What happened is that some time ago the W3C ratified - you're not going to believe this - a battery status API for HTML5. It's now in - it's been in Firefox since v16, and it's in Chrome and Opera, not in Safari or IE. So this is not an issue for them.

But, for example, Chrome browsers on Android, that would be a place where you would have a battery. The battery API allows a website to determine whether the phone is currently charging or not; the current battery level as a floating point value between zero

and 1.0, obviously for empty to full; if it's charging, the number of seconds remaining until it's expected to reach full charge; and, if it's discharging, the number of seconds remaining until it's expected to fully discharge.

Now, sure, you could see how that could be handy. A website could, if you were, like, going to start playing a video, it might check to see whether your phone is on a charger or not. And, if it's not, is there a remaining charge in the battery to watch the movie? And if not, it could say, hey, by the way, if you're going to watch this uninterrupted, you ought to plug the phone in. Now, that would be a little unnerving to us privacy-related people who would think, wait a minute, how does this website know that my phone is not plugged in, or how much charge my battery has? Maybe that's not a concern.

However, think about it. It's yet another thing which is not changing rapidly, or you could argue is changing in a predictable fashion, like, even whether it's charging or discharging, if it's not either fully charged - well, I guess it can't be fully discharged or you wouldn't be using it on the web. But if it's not fully charged, then it's going to be, this time in seconds is going to be ticking down as it charges, or up, or down, or, well, anyway, you know what I mean.

So it turns out that some researchers, two Princeton University researchers were just curious whether anyone had decided to leverage this for tracking. And of course we know the answer to that: Yes. They found, in the wild, ads which are running JavaScript, which are querying the battery status API and using it and, like, merging it with other tracking-related material in order to enhance the integrity of their tracking. And we talked a long ago about the Panopticlick site, where a whole bunch of browser headers are munged together, and you're sort of ranked, like, whether you're unique with your browser, or how many other people who are probably not you have been seen with the same fingerprint. It basically is a way of fingerprinting you without using explicit tracking technology like a cookie that was never really designed for tracking, but makes it drop-dead simple to do. Instead, they're using these sort of side channel approaches for locking onto us.

So now we have the battery status API as one more thing. And I think it's kind of cool that a website could help you out, maybe show your phone's battery gauge or use it in some clever fashion or remind you that it's really not the way lithium-ion cells want to be handled to be discharged fully. We see that you only have 5% charge left. If that's your habit, you should consider plugging your phone in more often, and the battery that is not interchangeable will last longer. So, but, yes, again, another example of something for good not necessarily being used the way its designers intended.

Okay. So fooling the Tesla's sensors. To me, this is, okay, interesting. Sort of maybe stick a pin in the map. But I don't think this is a big deal. Some Chinese researchers working with some people from the University of South Carolina demonstrated at DEF CON, using off-the-shelf radio sound, meaning ultrasonic, and light-emitting tools, the technology to both spoof the presence of nonexistent obstacles and mask the presence of real objects in the car's path. And so they're saying, you know, well, we're just demonstrating this academically. But imagine how bad guys could use it. And it's like, yeah, okay. To me, no one ever imagined that this technology was unspoofable. The car is doing the best it can. I'm still amazed that anyone is taking their eyes off the road or their hands off the wheel. I know you're not supposed to. But I'm amazed that this technology has just sort of exploded onto the scene as quickly as it has.

Leo: You have cruise control in your car; right?

Steve: Yeah.

Leo: Actually, I don't know what you have in your car. I don't even know what kind of car you drive.

Steve: I'm able to push a button, and it holds the speed that I'm currently going.

Leo: You wouldn't take your hands off the wheel if you had cruise control turned on. You'd keep your eye on the road.

Steve: Yes.

Leo: So it's three things. It's a smarter cruise control that adapts to maintain a good stopping distance, and you can set the stopping distance.

Steve: Nice.

Leo: So if the car in front of you slows down, you slow down. If it comes to a stop, you come to a stop.

Steve: To create a buffer zone.

Leo: Yeah. It's called "adaptive cruise control." My Audi did that. The second thing is it will stay in its lane. So if it can see lane markings, which it can't always, but most of the time on the highway it can, it will - and we talked about this a couple weeks ago. It stays right in the middle of that lane.

Steve: Right, right.

Leo: And so both of those, I think, are minor. My Audi didn't maintain, didn't steer, but it would warn you. It would pretension the seatbelts and vibrate the wheel. So that's not new. The one thing that's a little weird is you can turn on the blinker, and the car will change lanes for you.

Steve: That's nice.

Leo: But if you think about it, self-parking is similar; right?

Steve: Yup. And you know, when you mentioned the blinker, it's a pet peeve of mine that I'm a crazy blinker user, and I'm, like, alone.

Leo: You're the only one, I know, especially in Southern California.

Steve: No one, they just don't bother.

Leo: No. I was taught as a - because I learned how to drive when public schools still taught you how to drive. They don't anymore, by the way. There's no drivers training. But the drivers training teacher, and it stayed in my head my whole life, still does, said even if you're signaling as a convenience to the person behind you, give them some warning, even if it's obvious you're going to turn, or there's nobody behind you, just get in the habit.

Steve: Yes.

Leo: And I did. And I still have that habit. But a lot of people apparently weren't taught.

Steve: I think it's a courtesy.

Leo: Yeah. That's what he said, it's a courtesy. Let them know. And so, yeah, you can't change lanes without doing that on the Tesla, I guess.

Steve: It would be nice if we knew that the person blinking was going to do what they said they were going to do.

Leo: Right.

Steve: Because of course you still have to make sure that they're not going to keep on...

Leo: Well, as I get older, I realize, I'm just going to leave the blinker on. I feel like, why turn it off? You know what I'm saying?

Steve: No, Leo, you can use that emergency flasher button, and everything blinks.

Leo: I'm turning eventually.

Steve: Then you're covered.

Leo: I'm going to be turning. Just a little early.

Steve: You haven't decided which direction. Just flash them both.

Leo: I don't know. I'm not sure, yeah, mm-hmm.

Steve: That'll keep everybody confused. And they'll give you lots of margin, too, plenty of leeway.

Leo: I have that left turn signal going the whole time. The whole time. Just in, you know, you never know. No, actually the Tesla...

Steve: Let's take a break. I want to catch my breath.

Leo: Okay, yeah, we'll do that. I'll just, you know, the Tesla, I think you'd have to really be kind of strange and cocky and maybe overestimating what's going on. It's pretty apparent it's not really driving itself. And, man, I mean, every time I use it, and I do use it a lot, I'm paying just as much attention to what's going on. I'm just letting it help me.

Steve: I think not everyone is us.

Leo: Right.

Steve: And that's a problem.

Leo: We use blinkers. Right there we've established.

Steve: Right, yeah.

Leo: We're weird.

Steve: So Let's Encrypt. Everybody knows what a fan I am of the idea of automating the least verified class of certificates - which, while being least verified, still makes them incredibly useful - the so-called DV, the Domain Validation certificate for a web server, where the only thing it's claiming is that I am the server for this domain. There's no corporate association, no company reputation, nothing more than I'm a server on this domain. I mean, it's sort of obvious, when you think about it.

And so the beauty of what Let's Encrypt does is it allows those certificates to be made available in an automated fashion at no cost, thus moving the whole web from HTTP to HTTPS, to encrypt all of what was plaintext. So all kinds of problems. Like, I mean, looking back on it now, it's hard to, like, it's hard to believe that we were in a period during this podcast where you would log onto Facebook with a secure connection, and then it would then drop you back to a nonsecure connection, that is, a non-private, non-

encrypted connection, where your Facebook session was maintained by a cookie being sent in the clear.

Leo: Firesheep.

Steve: Yeah. It's like, did we ever actually do that?

Leo: Kind of amazing, isn't it.

Steve: Yes, the whole industry did that. So anyway, this is neat.

Leo: We've come a long way. We've learned.

Steve: We really have. We really have. And in fact, didn't we just hear that, what was it, it was like last week was the 25th anniversary of the first web page that was delivered probably at CERN by what's his name.

Leo: Tim Berners-Lee.

Steve: Yes, Tim Berners-Lee.

Leo: Sir Tim Berners-Lee.

Steve: Yeah, 25 years.

Leo: Can you believe that's the first web page. Wow.

Steve: And here we are closing in on 11 years of the podcast. So we've been around for a chunk of that time.

Leo: Wasn't it the 35th anniversary of the IBM PC, as well?

Steve: Yes. And by the way, I heard you ask who had a 5150.

Leo: Did you?

Steve: I had a [crosstalk]...

Leo: The original? Did you have the cassette port and the cassette adapter?

Steve: No. I came along, well, because I was in...

Leo: Because originally it did not have a hard drive or a - it had a cassette adapter; right?

Steve: It had dual floppies.

Leo: Dual floppies and a cassette adapter.

Steve: There was, yes, there was an audio, a microphone and earphone plug in the back so that you could store your 12-line BASIC program. I don't know who they thought was going to buy this thing with a cassette player. But, yeah. And so it had a choice of either a color or a monochrome display. The color display was - I don't know who designed it, but it flickered like crazy because, while the CRT is scanning, it's having to, in order to scan the screen, it's reading out of the memory for the video. And there wasn't enough bandwidth to the memory for the video to read from the refresh memory at the same time that the computer, the 4.77 MHz 8088 - or was it 8086? I don't remember on the very first one - when it was updating the video.

So you had a choice. You could either not give the video screen access to the video memory when the processor needed it, which would cause a little blerch to appear. And what you got was a screen full of static while the page was being updated. It would just sort of snow. And then IBM said, oh, no, no, it looks broken. We can't ship it that way. So some bright engineer said, oh, we'll turn off the video. We just won't let them see the snow.

So they blanked the video while the system is updating, like when it scrolls. In order to scroll the screen, you have to copy all of the data. You had an ASCII byte and then a color byte, so 16 bits per character. You'd have to copy them up by, in this case, 160 bytes, which is two times an 80-character line. So you'd have to copy this 4K buffer up by 160 characters, so basically move the entire buffer in order to scroll. Well, it looked like a blizzard. So they'd turn the video off, copy everything up, then turn it back on. Then what you got, if you did like a directory listing, it would just flash as it was scrolling upwards. So of course my first product for the IBM PC was called Flicker Free.

Leo: Ah.

Steve: It redesigned...

Leo: Vertical blank; right?

Steve: Yes, well, it did a nonblanking flicker. It turns out that something that apparently the IBM engineers hadn't noticed is that in the - it was a Motorola video display chip, so

it was a 68 something or other. There was a register that contained the starting address of the video memory. Well, they left it at zero. But it turns out, if you changed the starting address to 160, suddenly, with no flicker at all, the entire video buffer has just moved up, with no flicker.

Leo: Just you're jumping the buffer.

Steve: Well, instead of moving the buffer, I was moving...

Leo: The pointer, yeah.

Steve: ...the region that was being refreshed. And it got a little tricky at the end because the video memory wrapped around, yet it wasn't a multiple of the page size. So I had to do a bunch of other things. And, you know, nothing is ever as easy as the top-level description. But bottom line was not only did it eliminate the flicker; but, because you weren't having to copy the memory it scrolled instantly. And so people who were used to seeing their display go [vocalization], it just shot by. Anyway, so we sold a lot of Flicker Free. And that's what allowed me to then have time to write SpinRite.

Leo: That's very cool. I didn't know that, actually.

Steve: Yeah.

Leo: I thought I knew everything about you.

Steve: So, yeah, I had one of the big steel IBM boxes in the beginning. So anyway, Let's Encrypt is taking off like crazy. We've talked about what a success it has been. The problem initially has been who's going to trust their cert because they're needing to issue certificates on the fly using this confirmable API that allows a web server to automate the process of interacting with the automated certificate authority to issue its own certificates.

So what was done initially, and we discussed this at the time, was cross-signing the certificate, meaning that they did create, the Let's Encrypt people created their own root certificate, which upon launch no browsers knew about. So it didn't do them any good. But they also had their certificate - so the certificates they were issuing were all being signed by their own root cert that, again, nobody recognized, and also by an old-school major certificate authority that everybody knew, a process called cross-signing. So it was double-signed.

The news is that Firefox has just recently pushed the ISRG X1. ISRG - what is that, Internet Security Research Group, I think - X1 root cert is now in the - it's, like, through debugging. And I've got, if anyone's curious for details, links in the show notes to the dialogue where it's been accepted. Everything is called a bug even when it's not a bug. They just use the bug system to move updates through and manage it. So it's been managed. And I expect before long we will have a production version of Firefox that natively recognizes this ISRG X1 root cert.

And at some point - there's nothing wrong with cross-signing. At some point, after all the browsers have come up to speed and have that root cert in all of their stores, then Let's Encrypt can drop the signing by the other certificate authority and have it only signed by their own because theirs will be recognized. So another nice milestone for this terrific project. And LastPass I had to mention because a lot of our listeners sent me the news that they have produced an authenticator app. And so this is like...

Leo: Ugh. Ugh. Ugh. I'm sorry.

Steve: What?

Leo: You touched a button. First of all, they came out with that a while ago. But just this week they decided to pester the hell out of us about it. That's why everybody's sending you notes. Is it driving you crazy, too?

Steve: So, well, what's clever about it - so I should mention that it's the time-based one-time password that we...

Leo: Yeah. It's like Google Authenticator or Authy.

Steve: Just like Google Authenticator, like the old football that we talked about years ago. What's a little different, though, is that it knows about the LastPass browser extension, so that you don't have to type anything. When you use their Authenticator, and you're a LastPass user, which is probably the only reason you'd have the LastPass Authenticator, and your LastPass extension is logged into your browser, that is, when everything's, like, set up to go, and you visit a site that wants a six-digit code, the extension will see that. It will push a request to the phone. The Authenticator pops up, showing you the six-digits, but you don't have to enter anything. You simply say, yes, I want to use it to authenticate. And the six-digits are sent back to the browser app, which then populates the field on the site that you're visiting, and you're logged in.

Leo: And this, by the way, is exactly why I didn't use it. I don't want a single source for the authentication and the password manager. Because if one is compromised, then I'm giving them both; right? I didn't want to use the authenticator...

Steve: They're coming from a single manufacturer.

Leo: Well, if LastPass is compromised, then they get the password and the second factor. So that's why I don't use LastPass Authenticator. And I wish they'd stop bugging me about it. Because every time it pops up, saying, oh, you know, if you were using LastPass Authenticator. And there isn't even a close box on the popup. You have to go into a menu to close it. It's incredibly annoying. And I think it's less secure. I mean, maybe I'm wrong. But I don't think having a single provider do both is the right way to do it. I understand what you're saying. I'd still have to be

authenticated and logged in as LastPass. But let's say I've been compromised.

Steve: Well, okay. The reason I'm hesitating is I'm just sort of running through the logic because you still have an app running on a mobile device. And so that gives you...

Leo: No, it's not on your - it's in your browser.

Steve: No. The Authenticator is on your phone.

Leo: Oh, I misunderstood. Okay. So I did install this, by the way. Like a couple of months ago I installed the LastPass Authenticator. But the problem is you're setting up the Authenticator with your LastPass credentials; are you not? Maybe I misunderstood it. I just like the idea of using a separate company with a separate database for my second factor.

Steve: You're completely right, if the way it worked was that LastPass browser extension was the authenticating agent so that it populated the six digits. Then I'm 100% with you. But that's not what this is.

Leo: Okay, good. Because I use the Google, the new Google authentication, which sends a notification to your phone, and then you accept it.

Steve: This is that.

Leo: Okay.

Steve: That's all that they've done. They've done exactly the same thing.

Leo: Then I'll turn that on. You know what, maybe this is new because they did something a little different. When they first did it, it looked like another - it looked like basically another Google Authenticator, and it didn't have that wiring.

Steve: And you know, Leo, just to be honest, what you're reporting in terms of being bugged?

Leo: That bugs me.

Steve: That's not something Joe would have done.

Leo: It's a bad sign, no, unh-unh.

Steve: We never had that before.

Leo: An ad for another product popping up whenever I use LastPass? Unh-unh. That's a very LogMeIn kind of thing to do.

Steve: That's exactly my...

Leo: No, I'm not happy about that.

Steve: It does support time-based six-digit codes, what they're calling "one-tap push notification."

Leo: That's the thing I like. That must be neat, yes.

Steve: Yes.

Leo: I do like that because I use that with Google, and it really is convenient, much more convenient than entering the six-digit code. All right. Good.

Steve: We're getting so spoiled. I have to type those pesky six digits. I can only remember five. I have to keep going back and...

Leo: That's kind of part of my argument is it shouldn't be easy; right? It should be hard.

Steve: Oh, I've got something coming that's easy.

Leo: Okay. I guess SQRL. SQRL.

Steve: I've got something. So I wanted to tell our listeners that GRC's DNS Spoofability Test is back up. I didn't realize until I inadvertently killed it a couple months ago how many people used it. And it is very cool. And there's nothing else like it on the planet. And I'm proud of it. And it was a huge investment of time. And after I was done, I thought, well, it's beautiful, but why did I spend all this time? And I'm really hoping I don't feel that way about SQRL. So we'll see. But this thing, it died. And the reason I'm bringing it up, first of all, I just wanted to let everyone know it's back.

What happened is I use DNS in a very clever way for GRC's version checking. And the other thing, the DNS Benchmark, people who've used the DNS Benchmark all the time

started complaining. He says, "Hey, it says it can't tell if there's a new version." What happens is I have what I call a "pseudo DNS server" that I wrote at GRC. And so domain names can be special. For example, DNSbench.ver.grc.com looks like a domain name, and it returns an IP address which is actually the least significant two pieces of the most recent version of the application's version number. And what's nice about that is that it's not TCP. It's super low overhead. Anything lets you do DNS, even behind a portal or something. So it tends to get out. And all it's doing is apparently making a single DNS request. So it's just a beautiful lightweight way of checking versioning.

Well, I had to add that to SQRL, the GRC SQRL client. I mean, we're at that point now where I'm in the final details of this stuff. And so I knew that the DNS system had broken a few months ago because we were getting complaints about it. But it's like, yeah, I'm busy. Yeah, I'll get to it. But the complaints kept coming in. So I'm gratified that people are finding the DNS Spoofability Tests to be as useful as they are, and if it's getting use, that's neat. What happened was weird. And it put me in mind of that CUJO appliance we talked about a couple weeks ago because I inadvertently CUJO'd myself when I brought up the new FreeBSD Unix server, for a reason I don't yet know, even now. I know what the problem is. I don't know what's causing it. And it's bizarre. But it's claiming an IP that it doesn't own, which happens to be an IP that gets mapped internally to the incoming DNS responses, or the incoming DNS queries.

So DNS comes into a nameserver at GRC. And that Unix machine continues sending out what's called a gratuitous ARP. Now, normally the way address resolution protocol works is that, when something on an Ethernet needs to know what network adapter to send a packet to based on its IP - remember, Ethernet is the numbering, the labeling, the addressing is the word I'm looking for, the addressing on Ethernet is MAC addresses, these 48-bit MAC addresses, where the most significant 24 bits is a manufacturer number, and the least significant 24 is a serial number of that manufacturer, the idea being that you don't need to have jumpers or switches or anything. There will never be more than that many Ethernet adapters on the same network, certainly. But the idea is never in the world. So you just plug them onto an Ethernet, and you don't ever have to worry about address collision.

So ARP says - it's a broadcast message that says, "Who has IP X?" And so all the adapters on the so-called broadcast domain, which is typically the whole Ethernet, but that's something that VLANs are able to segment, all of the adapters that hear that, they check their little list of IPs. And you know how adapters can have more than one IP. This is how. This is how you can have multiple IPs on a single computer or adapter is they just have a list. And if somebody is asking who has this IP, the adapter that has been configured with that IP says, "I do." And so it responds. So there's an ARP request, and an ARP reply.

Well, the other thing that can happen is what is happening in this case, and that is a gratuitous ARP. As the name sort of implies, it's a statement that, like, nobody asked for, but the owner is just sort of saying it. And that's handy. So, like, when an interface comes up, when you're booting a system for the first time, and the networking system comes up, that NIC, the Network Interface Controller, can send gratuitous ARPs for all the IPs that it owns. And so what happens is any switches which are on the network will receive those gratuitous ARPs and go, oh, nice to know. So you save them the overhead of asking, when something wants to come in, that it sets up their ARP table automatically.

So what was happening was this UNIX machine is every second or two sending out a gratuitous ARP for an IP completely unassociated with it. I'm tempted to think this is a bug. And what's interesting is I have three Unix machines. The oldest one is behaving

itself and is not doing this. Both the one I set up recently to handle GRC's forthcoming web forums and the FreeNAS server, they're both doing it. The FreeNAS server wasn't causing problems. Even though it's also doing it, it's claiming an IP on the same switch port, so it doesn't cause the switch to send the traffic out the wrong port. So it's bad, but it's not disruptive. Which is why I'm thinking maybe there's a bug that nobody has, I mean, because you wouldn't notice it unless you were in this weird situation where the machine was claiming an IP on a different switch port, thus disrupting traffic that should have gone out that switch port, sending it over to its.

And the point is this is how the CUJO works. Remember we talked about it, this thing that you just plug into your network, and somehow it's able to take over, it's able to do a man-in-the-middle attack, essentially, benignly. But the idea is it sends out gratuitous ARPs in order to claim to be the gateway for the network. So all the devices send their traffic to it, rather than to the actual gateway. Anyway, so I thought that was a little interesting back story, and just wanted to let everybody know that the DNS Spoofability Test, GRC's DNS Spoofability Test, is back online. Oh, and the DNS Benchmark is happy again. It's able to see what its version is. And now I can write the code to add it to the SQRL client so it'll be able to let people know if I've got a new version of the SQRL client.

And one last, or actually two last bits. A follower, Justin Garrison, he has been following our conversation about, from a Q&A a couple weeks ago, the listener who was wondering about using Firefox as a noncorporate-intercepted proxied web browser. And he reminded me that the portable version of Firefox maintains its cert store on its portable medium. So if you install the portable version of Firefox, for example, on a USB drive, and if your corporation allows a nonproxied browser to have access, that's another way of avoiding anyone messing around with its certificate store because it's off on its own and is only present when you're using the browser.

So thank you, Justin, for the tip. And for anyone else who might find that useful, it's nice to have the option of something like that, a fully self-contained little portable Firefox instead of, for example, I was suggesting a VM as a means of creating containment. But this is probably even cleaner. And it's not something you probably need to have all the time and wouldn't take up any space on your system. And Leo.

Leo: Steve.

Steve: "Stranger Things."

Leo: Oh, you watched it? Did I tell you to watch it?

Steve: Yes.

Leo: Okay.

Steve: You mentioned it, well, you mentioned it kind of diffidently. And I just had to say...

Leo: Well, I don't like to tell you about shows if you don't like them. I don't know.

Steve: It's the best thing I've ever seen in my life.

Leo: Oh, my gosh. You liked it.

Steve: Which maybe is a little hyperbolic. So I will say - I was rehearsing this, and I just - it is one of the best things I have ever seen.

Leo: Good.

Steve: I absolutely loved it. It's a fabulous miniseries. I call it that because it's eight episodes. They're like an hour and 15 minutes each. I mean, Netflix has done it again. I just bit my tongue until halfway through Episode 7. I kept waiting to see if it was going to disappoint me, if it was going to fail. And finally I said, okay, I can't wait any longer. So I tweeted out an absolute, unreserved recommendation. I mean, it's worth joining Netflix for - and they've raised the price now to $10 - joining them for a month just to get this, and then quit your subscription if there's nothing else that you find interesting. It is really good.

Leo: It's interesting because it's had a slow start. It's like they didn't promote it or something. But the word of mouth has been spectacular.

Steve: Well, and you were onto it pretty quickly. It was July 15th that it was released, so only the middle of last month.

Leo: People are discovering it now, though.

Steve: Well, it's a 9.2, which I don't think I've ever seen on IMDB. It's a 90-something, maybe 92 or something in Rotten Tomatoes.

Leo: Good.

Steve: I consider it, when I was trying to, like, characterize it, I called it a cross between "Goonies," "Super 8," and "Close Encounters."

Leo: Exactly. It's self-consciously so because it's a very - it's an '80s homage.

Steve: Right. But also it starts out as sort of a mystery. It opens with a star field, and you slowly pan down to this dark, mysterious-looking laboratory with some satellite dishes and some red lights upon poles. And, you know, the music is well done.

Leo: Music's really well done. I love the music.

Steve: It's fabulous. But then the way it successively reveals what's going on. Also, there is always a problem that it's not going to hold together, that is, like the writers cheated or something impossible happens. That doesn't happen. So the whole, once you really understand what's going on, and you don't until very near the end, then you realize how everything fits exactly right. So it has integrity, internal integrity, as well. And a fabulous ending. So anyway, I don't know if there will be another one.

Leo: I think they did get renewed for a second season. I don't know what they'll do with it. I guess there are a lot of unresolved questions that they could address.

Steve: We did have a little tease at the very, very last scene. So they could do more. Anyway, I loved it. And I know that not everybody follows me on Twitter. So Leo, thank you for mentioning it.

Leo: Good.

Steve: I did, I got around to it, and I stretched it out over two nights.

Leo: Yeah, I think we did - we did a few more. Because that's a lot.

Steve: Yeah.

Leo: It's like six or eight episodes, and they're...

Steve: It's eight, yeah.

Leo: Yeah. But it's hard to stop watching.

Steve: And I will say, the responses to the tweet were people who were already ahead of me said they absolutely agreed. And then others whom my tweet incented came back a few days later and said, wow. So for what it's worth.

Leo: Great cast. And there's kids, teenagers, and adults. And they each have their own threads. But the kids are the best. They're so good.

Steve: Oh, like, it's just, it's perfect.

Leo: So good, I know.

Steve: The dialogue, I kept backing up and, like, watching scenes over a couple times because - and I sound like Andy going off on something [crosstalk].

Leo: If you like Dungeons and Dragons, the Dungeons and Dragons scene, but beware the Gorgon. That was my only negative was the Gorgon. Little rubbery. Little rubbery.

Steve: Yeah, I agree, I agree.

Leo: I'm trying to do this without spoiling anything.

Steve: Also the dumb adults versus the smart kids, you know, the parents are kind of clueless.

Leo: Yeah. Yeah, poor Winona Ryder didn't really, I mean, all she could do was act freaked out for eight episodes.

Steve: And I wouldn't have even recognized her.

Leo: I know. I know.

Steve: I was very surprised. And I knew from the credits. But it's like, wow, that's Winona?

Leo: "Stranger Things." It's a Netflix exclusive. You have to have a Netflix account. I think there's enough good stuff on Netflix that it's worth the subscription, 10 bucks a month. And by the way, because I have now, I bought a UHD HDR 4K TV, that's the other good thing about Netflix is they have a significant amount of 4K content, and even some HDR content. And you can find it by searching HDR on Netflix. And so even though "Marco Polo" was the worst series ever, it sure looks good. It's beautiful. Anyway, "Stranger Things," Netflix.

Steve: Yup.

Leo: Thank you. Thank you, yeah.

Steve: I don't think anybody could be disappointed. I mean, yeah, there are curmudgeons. But still, if you're not a curmudgeon, if you like the idea of something fun...

Leo: Well, it's funny, you know, I don't want to recommend something to you that you hate. So I'm always, like, a little ginger in, you know...

Steve: I can certainly hit pause or stop.

Leo: [Crosstalk] something that you kind of like.

Steve: But anyway, I'm delighted.

Leo: All right, Steve.

Steve: So I wanted to mention that I did bring myself up to speed on what this BeyondRAID is.

Leo: Ah, I was very interested to hear about this, yeah.

Steve: And also what they mean when they say "thin provisioning." First of all, the thin provisioning is a - essentially it means, when you create multiple volumes, you don't need to pre-declare the size of each volume. In the old days we had hard drives with, well, we still have hard drives, fortunately, and partitions. So you can think of a partition by the actual name it's been given, a partition of the hard drive. And you would set things up with C, D, E, and F partitions. That's like the reverse of thin provisioning because the point is that you have to decide ahead of time how much space to give to C, how much to give to D, and so forth.

Now, things have gotten better. Windows, for example, is able to shrink and move partitions, which allows you to make some changes. But what the Drobo does is allows you to - and what they call "thin partitioning" is to create multiple accounts which are able to grow at whatever rate they want to grow and not need to preassign any specific amount of space to any one, but essentially allow them to all pull as they need to from that space. So that's their thin partitioning thing.

But the technology that was most interesting to me, that I wanted to understand, was what they called BeyondRAID. And what I like about this is that what RAID itself, the abbreviation, Redundant Array of Inexpensive Drives, RAID, the concept is that you are protected against the failure, either a spot can't read or the whole drive can't read, but you're protected against failures by redundantly having information on the drive.

So let's take the three drive case. You have two drives that contain different data. So say that you had a 10MB drive and another - okay, I really am old school, forget about that - 10GB drive, another 10GB drive, and they've got different data. They have different data, so it's not - the data is not in any way related. Now you take a third drive, and what's really cool about the XOR function is you XOR the data, sector for sector, byte for byte, from those two drives, and you store the XOR of them in the third drive.

So what this allows - so now you have a three-drive RAID. Now, its still only stores 20GB, right, because we have two drives, 10GB each. And then we added another drive.

But what it's doing is storing the XOR of the first two. So that doesn't give you - that doesn't buy you or deliver any more storage space. It still, the total storage is still the two drive sizes, this 20GB. But any one of them can fail, and you lose nothing. Obviously, that third one, that checksum drive that we added, the XOR drive, that can go away, and no one cares.

What's cool is, since it stores, essentially, the difference, that's what an XOR is, the difference between the bits in the first and the second drive - and I'm not going to take that drive away, I'll take that drive away - if the first drive goes away, we can infer the contents of the first drive by XORing, again, the second drive against the XOR drive. And that returns the data that was originally stored in the first drive. And what's elegant is that, from a computer science standpoint, XOR is very fast. It's incredibly, well, not incredibly, but it's sufficiently fast that you're not losing a huge amount of speed in essentially arranging to - so that whenever you write data to either drive, you update this XOR sort of extra copy that represents the difference.

Now, what's really cool is that it works with more than just two drives. You can take three drives as your main, or four drives as your main, or five, and then have one XOR of all the others. And if any one of the others fail, you can reconstruct what it originally had from using the remaining drives. You'd need, I mean, it's going to be slower because you have to read all of them in order to recompute the missing drive's contents, but it works. Now, in an extension to that is RAID 6. And I can't do that with my fingers, unfortunately, because it uses some fancier math. But essentially it's an additional drive of redundancy which also does not buy you more storage. But now it means any two drives can fail, and you still can read the whole thing.

And RAID 6 is coming into its own because drives are now - drive arrays are getting so big that, if a drive dies, in RAID 5 if a drive dies, you have lost all redundancy. It's okay, as long as no other drive fails. But the concern now is sometimes it takes days to rebuild a failed RAID when one drive dies because, as I said, it is intensive. You have to read the data on all the other drives in order to reconstruct the one missing drive in a RAID 5.

And so if you've got 12 drives, for example, and those are multi-terabyte drives, as I've mentioned before, nobody ever formats a drive any longer. I mean, if you actually gave a format command where it was going to go out and read every sector, you might as well just, I don't know, come up with some way of needing less space because this thing will just never happen. It'll never finish. That's the burden that SpinRite has because it's all about actually reading all of the physical surface. Nothing does that any longer except SpinRite. But that's also what makes it unique and valuable.

The point, though, is if you have 12 drives, one of the problems with increasing the drive count is if each drive has a certain probability of failure. The more of those you have, the more likely one of them will fail, statistically. So the problem with RAID 5 is that what happens if another drive fails during the rebuild of the first drive that failed? That's why RAID 6 is gaining popularity. Drives are cheap, and technology is cheap. So let's add one more drive. It won't give us any more space, but it will mean that, while rebuilding from a failed drive, if anything else fails, we're still okay.

So that's sort of the RAID background. When you think about it, all that's really necessary is that we arrange for redundancy across drives. And what I was curious about from just the overview of what Drobo did is it made it sound like you could remove a drive, and say that you had it populated with five, as I do mine, you remove a drive. And as we know, if that was just RAID 5, standard RAID, I remove a drive, and I have now lost all my redundancy because I'm now requiring all of the drives I have to compute the data in the missing drive. And so that's a problem.

What Drobo has done is they said, okay, so we're down X terabytes from the drive that's failed. So now we've got four. But rather than saying, so we have no redundancy, they say, we're going to restore redundancy as quickly as possible. And when you think about it, nothing actually prevents you, theoretically, from re-RAIDing on the fly, that is, saying to yourself, okay, we only have four drives. We want to restore redundancy. So let's turn this into a four-drive RAID. And they do that. Which is, like, very cool.

So what they do is, when they see a drive is gone, they start rebuilding to restore redundancy. Now, the requirement is free space because what we've seen is that the way this works is you are consuming a whole drive's worth of space for redundancy. And the other thing to note, in a three-drive RAID, you've got two drives of data and one of overhead. So that's not very efficient. One third of your total capacity is wasted in this RAID checksum, you know, the XOR drive.

So what is nice with a simple RAID 5 is the more drives you have, the lower the overhead becomes of the redundancy. Because, as I said, if you had like 11 drives - and I don't have that many fingers - 11 drives, and then one of checksum, then it's now much lower percentage of the total storage. On the other hand, as we saw, with that many drives the chances of any of them having a problem goes up. So then you begin to want to have additional redundancy.

Anyway, so all I wanted to really say was that I'm impressed. The way they pull off this, you know, and we mentioned it before - in fact, last week I talked about a guy who was using SpinRite to recover friends' drives and then copy them to brand new drives. And then he would take that recovered drive home with him. And if it was bigger than the smallest of the drives in his Drobo, he'd pull out the small drive and put in the bigger drive.

And again, what Drobo does is it has the technology, first of all, unlike other RAIDs, the drives contain metadata that tell Drobo which drive is which. So you're free to rearrange them in the box anytime you want. If I did that on my hardware RAID at Level 3, everything would fall apart. It would not recognize the RAID, and it would say it was broken, and alarms would go off. Drobo doesn't do that because it tags the drives and tracks them as they move around within the case. But in this case, so you pull out the smaller one, the smallest one, put in one that's larger than that. Drobo recognizes what has happened and then sets about rebuilding itself to optimize the storage that it contains.

And you do have a choice of one or two drives of redundancy, that is, sort of a drive's worth of redundancy. So you can run where, for example, in a five-drive box, where you get the storage of four of the drives, if you use a single-drive redundancy, or you can do dual-drive redundancy and be protected if any two die and then have the storage equivalent of three of those drives. So anyway, I just - I did want to dig into it. I hadn't seen this technology before. They did it right. And I'm very impressed. So, nice piece of work. SpinRite.

Leo: Speaking of nice pieces of work, yes.

Steve: I found a nice question from somebody who was wondering about ZFS, the Z File System. Trevor Harrison in Vancouver, British Columbia, the subject was "With ZFS Scrub, is SpinRite still needed on ZFS volumes?" He says: "Hi, Steve. I'm building my first FreeNAS box. Hanging out in the FreeNAS IRC channel, I've been told that the ZFS

Scrub is a block-by-block scrub of the drive to find bad blocks. I know SpinRite works at the sector level below the file system." He says, parens, "(At the metal level?). So can you explain to us, and even do a ZFS in-depth podcast, as to why ZFS Scrub replaces SpinRite or doesn't? I personally think it doesn't, but running SpinRite on a live server with lots of drives, well, you get the idea. I'd like to know for sure and understand the differences." And he says: "Do you ever SpinRite your GRC servers?" And then he says, "Also, I'll be buying SpinRite as soon as I can hear my 'Yabba Dabba Doo' live. Kind regards, Trevor."

Leo: I told you you should have that live. I told you.

Steve: Yeah. I've had a lot of requests for that. So I think in the future I'm not going to mute all of the Yabba Dabbas.

Leo: Oh, that's fun. Mute it when it gets out of control. And let's hope it does.

Steve: Because that would be fine, too. So it's not very disruptive. Okay. So here's the deal. ZFS Scrub is not what it sounds like. It's a nice term, and it wouldn't confuse you unless you were thinking, wondering whether it does what SpinRite does. It doesn't. What the Z File System contains that other file systems lack, and which is where it gets its reputation, its deserved reputation for robustness, is it goes checksum crazy. It checksums everything. And other file systems don't do that. There are ways to rebuild them, but they don't tolerate the overhead of checksums. ZFS was designed with this from the beginning.

So what a scrub is - and a scrub is not something that the file system does automatically, it's a command you can invoke. It can run in the background while the file system is live. But it's tying up a lot of resources, as we'll see in a second. So you may notice degraded file system performance during the scrub, and it can take a long time. I saw just - I saw like a 28-hour scrub report when I was poking around, learning more about ZFS a couple weeks ago. So, I mean, it's a multi-day sort of process, again, because file systems tend to be big now. And what the scrub does is it is a manually invoked checksum verify. So there's no reason to believe that the checksums would be wrong.

But the system doesn't just go through and scrub them all, which is the term ZFS uses for going through, without any reason to believe there's a problem, just to do - it's like a high-end verify pass. I want to verify that my file system is pristine, that no bit rot, as it's called, has crept in, because that can happen, as we've seen. And so this does that. This reads every block, computes the checksum of that block, and then verifies it against the stored checksum for the block, basically doing a checksum of the entire file system. So as you can see, that's not what SpinRite does.

SpinRite, of course, is about - oh, and it doesn't, obviously, mess with non-file system space. So it's not concerning itself with preserving or checking the integrity of any sectors not in the file system, that is, they're not in use. SpinRite is about the lower level, as Trevor says, the "metal level," of going through and absolutely verifying and/or repairing the data that the drives store. And to answer his final question, yes. Everybody who's ever worked at GRC, as far as I know, and I know I've checked in with several of them years afterwards, we all run SpinRite on any system we're setting up. I ran SpinRite on all of the drives in my new Win7 box. I run SpinRite on the drives at GRC. I know that Greg is running it all the time on any systems that he's setting up for others

and for himself.

So those of us at GRC are believers. And I just want to verify the drive because, without running SpinRite, you just don't know what condition the drive is in. And sometimes you're surprised. I've returned brand new drives where I've run SpinRite on them fresh out of the box, and it has pushed down the relocated sector count, or the seek count has gone way higher than it should. Both are things that SpinRite shows you while it's running. And that just means this is a sick drive. I mean, it's sick right from the box. And so I send it back, and I get another one. I'll run SpinRite on it, and it'll behave itself perfectly. And so I'm glad I didn't use that drive that was sooner or later, and probably sooner, going to give me a problem So, yeah, we absolutely use it for preventative maintenance.

And again, I plan at some point to talk about ZFS in detail because it is a fascinating, beautiful piece of work. And I'm seeing it more and more. In fact, I configured that most recently built Unix box - not the FreeNAS server. That is running ZFS just because that's the way it came, and it was easy to set up a ZFS "pool," as they're called. But the forum server is using UFS and a hardware RAID. I just - I want the experience of using ZFS with JBOD, Just a Bunch of Disks. And so it's the last thing I need to do before I bring that server public because I'll have to take it down in order to convert the file system over. But I want to do that because, everywhere I look, ZFS is taking over.

Leo: Good, because I'm ready to run it. I'm actually going to - I'm thinking about converting my home cloud, which is right now running Sandstorm - FreeNAS supports Docker; right? So I could run Sandstorm in a Docker on FreeNAS. And then I have ZFS. I'm thinking about doing that. It would be crazy to do that, but I'm thinking...

Steve: Well, you're having fun, and that's what...

Leo: That's all that counts.

Steve: That's all that counts.

Leo: I crave ZFS.

Steve: So, okay. HTTP protocol flaws.

Leo: And we're ready. This is now the Black Hat/DEF CON segment.

Steve: Yes. We're back to Black Hat and DEF CON. We talked - I'll do HTTP first because this is interesting. This is a new attack that's been called HEIST, H-E-I-S-T, which is an acronym or an abbreviation for HTTP Encrypted Information can be - we're skipping the "can be" - Stolen through TCP windows. There are some details of TCP protocol that I've never bothered to get into because it's really down in the weeds. Oh, I guess I did actually talk about it, years back.

The TCP window is something which is an evolution of TCP which is a agreement by the endpoints about how much received buffer each one has, which gives the sender permission to send unacknowledged data ahead. So each end is essentially, as the ACK packets are being sent to the other end to affirmatively acknowledge how much of the TCP stream has been received so far, and it's now possible to be a lazy ACKer, where you only acknowledge every while, as long as you do it often enough that you're not worried about freaking out the sender and causing it to start retransmitting packets that you've already received. So you want to acknowledge frequently enough that you keep everything flowing.

But the idea is that, especially when we have what's called a large bandwidth delay product, that is, we're at distant points of the Earth, and the bandwidth is low so that the delay is high. And what that means is the roundtrip time is extremely long. So TCP works by the sender sending something, and the receiver acknowledging its receipt. Well, since the acknowledgment can take a long time to get back, we want the sender to be able to send ahead. And so what happens is the acknowledgment packet contains permission, a 16-bit permission. And in later versions of TCP that's scalable. So you can not do it by bytes, but you can do it by larger increments because these days the bandwidth delay products are much bigger, or can be.

That permits the sender to - essentially that so-called "window," which is called an "advertisement," the window advertisement that the recipient is sending with its acknowledgment says, "At this time you are free to send this many bytes unacknowledged," meaning that the receiver can confirm its ability to provide the buffer space necessary. So the sender just can blast away without requiring incremental receipts.

So it turns out that what these researchers found was - and this is what's unique about this - a browser-based way of sensing the actual size of data at the packet level. Which sounds like, okay, who cares? Except that we've seen places, CRIME was one, BEAST was one, there have been a number of attacks on HTTP where, as we'll remember, where the compression was reverse-engineered. That is, it was possible by guessing a whole bunch of times to determine what some unseen compressed data was by putting your own data in, which would be combined with the compressed data. Remember that the way the compression works is that you get more compression if you have more redundancy. And so if you put in data which turns out to be redundant, the total size increases less than if you put in data that was not redundant.

So that's a really roundabout, indirect way of figuring out, painfully, but you know hackers, they're clever and patient, figuring out what data you can't see by seeing how large the compressed result of what you can't see, plus what you can see because you put it in, ends up being. These guys figured out a way for JavaScript to determine the size of data being sent. Which moves these previous man-in-the-middle attacks, because you had to be typically a man in the middle to make this practical, that's moved those attacks into the browser. It's still fringe, and not something you need to really run around worrying about, I think. But an interesting hack, and no doubt a great presentation for, I don't know, I didn't note here whether it was Black Hat or - oh, yeah, it's a Black Hat presentation. So that's HTTP, the HEIST attack.

The other presentation at Black Hat last week was four new, what I would call a "protocol implementation flaw," not necessarily a flaw in the protocol, but side effects of first Version 1 implementations in HTTP/2. HTTP/2, we did a podcast on it, I don't know, about a year ago, looking at what features it offered. There's a bunch of cool things it does. And this was originally driven by Google's efforts to improve the efficiency of the whole Internet.

An observation, for example, was that web browsers are almost always sending non-varying headers. The same, you know, if a web page is queried, all these requests go out for all the assets that populate the HTML page, and they've all got all these query headers, this metadata that's not part of the actual data goes out, you know, things like I have a copy that expires on this date. Do you, server, have something newer? I'm this user agent, I'm this language, I'm this quality of service and so forth, a whole bunch of stuff per query. Which typically does not change.

So it's been observed that this is a lot of redundant junk, especially if the little objects you're getting back are small. All of this header information can be a substantial percentage of the total payload. So reducing header redundancy, compressing headers. There's also this notion of streaming, where normally a browser is only allowed to have two connections to a remote server, to a given remote server IP, in order just to keep this under control. The problem is that a page which is loading a whole bunch of things, like maybe that page has a whole bunch of little things and a few big blobs. If the requests are made for the blobs, then all the little things have to wait in line behind the blobs, waiting for those to finally get sent by the server in order for the page to then be able to show all the little things that may actually be more interesting, or more immediately useful than the big blobs.

So what HTTP/2 does is it multiplexes that single connection. In fact, we do, we drop to one connection now, which is multiplexed, so that multiple queries can be sent, and multiple pieces of responses can be received in any order that the server wants to send them. And so that's potentially a huge win for just the feel of web pages.

Well, security researchers being what they are, they said, huh. I wonder what mischief we can get up to with this stuff? There is a famous old attack that we talked about a long time ago called a compression layer attack. It's also been known as a "zip bomb." The idea is, we've talked a lot about interpretation recently, about how, for example, the TIFF image and JPEGs and all kinds of file formats are sort of a meta description of their contents, so that the thing that is reading them is reading the file metadata, which then tells it what the following data is.

Well, compression is very much like that. When you have a compressed file, you have control information mixed in with data. And the presumption made by the decompressor is that a friendly compressor generated the compressed data. But that's not necessarily so. And it turns out that you can horribly crash HTTP/2 servers if you deliberately mess around with the header compression. It is possible for a single connection to tie up as much as a gig of server RAM. And you can do that as many times as you want to until the server crashes. So in their presentation of what they called the "HPACK Bomb," the attacker crafts small and apparently innocent messages which decompress into significant amounts of data, as in gigabytes, on the server, bloating the server's memory footprint and often crashing the server.

And so, for example, just to give you an example, one of the ways you might compress a block of null space, a block of all zeroes, say that you had 16K of zeroes. And in fact EXE files, one of the things that's so annoying about the Microsoft EXE format is that it's incredibly inefficient. And there are huge regions, huge tracts of land of null space in the middle of these files. So instead, an intelligent compressor recognizes, it sees all this null space and goes, wait a minute. And so it puts in a special marker that says, here's a zero. Duplicate that 16,000 times. Well, what if you said 16 billion times? Oops. So clearly we need sanity checking on the decompression to make sure that this compressed meta representation of the headers is sane, that it makes sense, that it's not totally crazy. We don't have that today. So that's an example.

Another is something known as the Slow Read attack, where a malicious client deliberately reads responses very slowly. And we did talk about this also, a long time ago. It was called the Slowloris attack. it was an early form of DDoS where you would establish connections with the server, and you just would not acknowledge the receipt of data very quickly, which forced the server to keep the connection up. And that allowed you plenty of bandwidth and time to initiate many, many, many, many, many, many, many more connections, all which would also be slow. And eventually they would pile up.

Well, it turns out that - this was done by Imperva Research, and they tested variants of essentially this Slow Read attack on Apache, IIS, Jetty, Nginx, and Nghttp2, all able to bring the server to a standstill at this point. Again, just something that the protocol doesn't yet handle, but needs to, in sort of version 0.9.

There's also some details in the way the HTTP/2 protocol is implemented, something known as the Dependency Cycle Attack, which takes advantage of some flow control mechanisms which are new in HTTP/2, which support network optimization. And in this case, again, a malicious client is able to craft requests that induce sort of an interflow dependency which puts the server into an infinite loop, essentially, as it tries to resolve these deliberately intertwined dependencies for which there's no resolution, sort of the Kobayashi Maru attack on HTTP/2. And then they've also come up with a way of abusing the stream multiplexing that I had mentioned before.

So these are not showstoppers. It doesn't mean that there's, like, we need to fix HTTP/2. That's why I call these "implementation errors." And we do need to fix the implementation. And I'm sure, after this presentation at Black Hat, all of the server implementations are in the process of playing with these attacks themselves and coming up with mitigations for them. So they all look like things like, oh, you know, we just didn't consider that a client would be evil. So with a new protocol you are inherently opening new opportunities for clients to misbehave. And that's why we have Black Hat and DEF CON.

Leo: Your mention of sanity check reminded me, if you don't mind a little interruption.

Steve: Yeah. Yeah, yeah.

Leo: I got an email from Patrick, our engineer over here. He runs the API and so forth. And there's somebody in Australia who loves this show because he has run a script to download every episode. A lot of people do that. But he neglected one minor detail. He forgot to put an end number in it for the last episode. So he has been downloading since July, the end of July, just downloaded, or tried to, Episode 306,003. Which I'm sorry to say we won't actually be recording until the year, according to Patrick's calculation, 7889.

Steve: Yeah. Your lease will have long since expired.

Leo: So just a little tip. If you're going to write a script to download, you might, A, want to keep an eye on it; B, have an upper limit. You know? In fact, Patrick suggests using our API because there are shows with weird numbers like 85A or

103SE.

Steve: Yeah, actually Security Now! has a couple of those.

Leo: Well, that's, yes, this guy's doing Security Now!. And the API will deliver you, not only all of the numbers, but not just a simple numeric succession, but in fact every show episode. And you can ask the API. And then you won't be downloading episodes in the high hundred thousands.

Steve: Which demonstrates a lot of hope.

Leo: There's a lot of hope. He doesn't want to miss any of them. You've just got to keep running that till you get there. Anyway, I'm sorry. I didn't mean to interrupt. But you reminded me.

Steve: No, it's okay. I've got one last item. And I was going to follow up on it anyway next week. So I'll just tease everybody with it. I need - this is bad enough that I need to understand it more than I do. And so I'm going to have to do a bit of research. But here's the deal. Back in 1997, 19 years ago, hackers found that Windows 95 and Win NT could be induced to send the username and password out of the system.

Leo: I know where you're going with this one. I know where you're going here.

Steve: And the way that was done is the web browser could be given, by a malicious web page, be given a resource on what looks like a Windows file share, a so-called SMB, Server Message Block, also known as CIFS. And that's the standard Windows file and printer sharing protocol. In the same way that most resources or assets that a browser queries are HTTP:// or sometimes even FTP or other things, you can do SMB. You can actually tell a Microsoft browser to get something over SMB that's not local. And so 19 years ago this was brought to Microsoft's attention. They know about it, knew about it, and said, yeah. Okay. We're not going to, you know, that's by design. We don't think it's a problem.

Now, the argument at the time was that somebody in Russia gets your PC's username and password. What are they going to do? Presumably, there's no way to log in remotely into that machine. So the fact that - oh, I should mention they don't get the password. Oh, and I forgot to say they get what is - oh, I completely missed the whole punchline. And that is that part of this SMB protocol provides your current logged-in credentials with the query. So with the query that goes out over the Internet from your Microsoft web browser, now, that's one of the mitigators here is that Chrome and Firefox don't do this. Only IE and Edge do. But Edge does, and why this suddenly got important is, as we know, Microsoft started using your Windows 10 authentication for much more than just Windows.

So whereas 19 years ago this may not have been a problem or, I mean, still would never make anyone comfortable who knew about it, suddenly it's a huge exploit vector. Any Windows 10 user who was using Edge - and I imagine that's what most Windows 10

users are using, unless you're a higher-end user, you're a listener of the podcast, and you've chosen to use Chrome or Firefox. But if you're using Edge, and you haven't done anything different, and you go to a web page that wants to get up to some mischief, that server will send your browser a request for something that it apparently has, that is, the server, remote server has, over the SMB protocol. Your browser will send your currently logged-in username and a hash of your password - you heard that right, your password hash - to whomever asked for it, wherever they may be.

There are currently two test sites that I'm afraid to go to. One of them is in Russia, and I'm sorry, but I'm not going there. But there are two. And people who have written about this have gone there and have had their password cracked in four seconds. So their username and their password for their Microsoft Windows Live account, cracked and known by an external party in a matter of seconds, just by visiting a web page.

Now, there are probably other mitigators here, and I just want to verify that. So I'll come back to this next week. For example, as I've mentioned, many people are now behind ISPs that filter ports 137 through 139 and 445. Those are the Windows file and printer sharing ports. So I'm assuming ISP blocking of Windows shares would solve this problem. And I, of course, with ShieldsUP, back in the beginning, I used to greet people by their name. I'd say, "Hi, Leo." In fact, that's what happened the first time Kate showed you ShieldsUP. I don't know if you [crosstalk].

Leo: It said my name, I did, and I was shocked.

Steve: So this kind of information leakage is old news, but it never went away. And I think Microsoft, it didn't occur to Microsoft that reusing your single sign-on credentials for all these other services might create a new opportunity for exploitation. There are some registry - and unfortunately that's where they are, in the registry. It's NT LAN Manager, NTLM keys in the registry that can be changed to turn off sending your credentials to remote sites. So the switch is there, but they're all on by default. And I loved it because I went, I dug back in, and I found the original report at Insecure.org. They were calling them "sploits," as they once did.

Leo: Sploit.

Steve: Yeah, sploits. And this was WinNT/Win95 Automatic Authentication Vulnerability. Here's what I love. It says, parens: "(IE Bug #4)."

Leo: Wow.

Steve: There was a actually a single-digit bug count at one point.

Leo: Wow, the fourth bug.

Steve: Bug #4. Leo, that's the fourth bug.

Leo: That must have been before it was released.

Steve: It turns out it's not a bug. Microsoft says, quote...

Leo: They did it on purpose.

Steve: "We're aware of this information-gathering technique, which was previously described in a paper in 2015." That is, it came back around again last year. And they say: "Microsoft released guidance to help protect customers; and, if needed, we'll take additional steps." Otherwise, eh. Basically they're saying, eh, we don't think it's a problem. So various sites are reporting on this. Our friend Lawrence Abrams at BleepingComputer has a very nice extensive write-up. He's one of the people who took a sacrificial machine and poked Russia, and they hacked his password. I mean, they decrypted his hash in four seconds.

And so if anyone is concerned about this, again, only IE and Edge. So, for example, as a Firefox user, I've never been in danger. As a Chrome user, nobody would be. But as we know, there are occasions where a link will explicitly bring up an IE or an Edge, or sometimes where something requires you to. Or you could even imagine somebody who was malicious putting up a web page that said this valuable content you want can only be viewed under a Microsoft browser. Please come back under IE and Edge. And maybe, if you wanted whatever you thought was there badly enough, you would. And then they could snag your username and your password hash. So that seems really bad. I'll have all the details next week.

Leo: Yeah, it's been getting a lot of attention.

Steve: Yeah.

Leo: I would guess that's probably the biggest story to come out of DEF CON or Black Hat.

Steve: Well, and 19 years old.

Leo: Pretty impressive, yeah.

Steve: And Microsoft, yeah, yeah, we don't think that's a problem. Uh, maybe it is.

Leo: Well, I mean, what are they going to say? Oh, yeah, it's been there for 19 years, and we didn't fix it? I think they're going to say, well, we didn't fix it because it's supposed to be there.

Steve: Yeah, I haven't...

Leo: So don't use IE or Edge. That's the key.

Steve: Correct. Correct.

Leo: You shouldn't be anyway.

Steve: Or Windows.

Leo: Or Windows, for that matter. I think Microsoft makes you use either Edge or IE for downloads. There are certain things you [crosstalk].

Steve: Oh, yeah, yeah, that's a very good point. That's a very good point. I run across that snarky text. "You must use Internet Explorer to obtain this." It's like, okay.

Leo: And I have installed, while you were talking, LastPass Authenticator. So we'll see. I'm not using it as my authenticator program, just for that one tap authentication. I don't know how that works, though. And we'll find out. Does it work with everything? Or just some things? Or I don't know.

Steve: LastPass needs to understand the site. So it needs to...

Leo: Right, know that it's asking for a code.

Steve: Correct. It needs to see that there's a query.

Leo: And it would then also need all the authenticator codes; right? You'd have to set up all the...

Steve: No.

Leo: No, you don't. Okay.

Steve: Those are in your phone. And so that's the key.

Leo: They are. Well, I'm saying you need to set that up in the phone, though; right?

Steve: Correct.

Leo: Yeah, yeah.

Steve: Yeah, and you know, I've been disappointed that the various authenticators do not have a universal export format.

Leo: Well, that's why I use Authy. Authy does, but the problem is it stores it on its Authy servers, which is...

Steve: Yeah, yeah.

Leo: It's encrypted. It seems to be Trust No One because I use my own encryption passphrase. But nevertheless. I don't know. Security's hard. That's why I listen to this show.

Steve: "Security is hard." Ooh, that'd be a good title, yeah. I think actually we've used that title several times.

Leo: "You're doing it wrong" is my favorite. Steve Gibson's doing it right, every week, right here, every Tuesday at 1:30 Pacific, 4:30 Eastern, 20:30 UTC. You can watch it live if you want at TWiT.tv, or you can download an episode after the fact. Steve's got them all at GRC.com. He also has transcripts, if you like to read along while you listen. Sometimes that helps with comprehension. Or it's just a great way to search. He's got lots of stuff at the site, while you're there. The sleep formula, all the stuff.

And SpinRite, the world's best hard drive maintenance and recovery utility. You could keep up on SQRL as we edge closer to the release day. And you can send him a little yabba dabba doo, while you're at it. Buy a copy of SpinRite. We have audio and video at our site, TWiT.tv/sn. So again, you don't have to be here live. You can always listen after the fact, or watch even. And every podcatch, I mean, 11 years, Steve, 11 years. That's a long time.

Steve: I thought doing the column for InfoWorld for eight years was a long time. But, yeah. I've passed it by three now.

Leo: It's remarkable. Well done. You can listen to every one of them. They're all there. Every one of them, if you really wish, 572 episodes, soon 573. We'll see you next time.

Steve: Thank you, my friend.