Android StrandHogg

Description: This week we revisit free upgrades from Win7 or 8 to 10, which can still be done; an alert for users of HP SSDs; the complications which arise with international privacy treaties when end-to-end encryption might be threatened; the U.S. government's formal permission to hack; a quick look at a particularly devastating ransomware attack; more anti-tracking privacy happiness coming soon, by default, to Firefox; the never-ending headaches caused by Windows DLLs; an update on my "Joy of Sync" determinations; and a look at the way some Android multitasking features can and are being actively abused - with Google's knowledge.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-743.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-743-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here. We'll talk about upgrading to Windows 10. It may not cost you a penny. A wonderfully written new memo from the Department of Homeland Security revealing a new vulnerability disclosure policy, and I think the best programmer comment I've ever seen. That and StrandHogg. It's all coming up next. You know what "StrandHogg" means? You'll find out on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 743, recorded Tuesday, December 3rd, 2019: Android StrandHogg.

It's time for Security Now!, the show where we cover your security and privacy and safety and how computers work and whatever, and science fiction, whatever else this man here wants to talk about because he is the star, and I mean star of our show, Steve Gibson. Hello, Steve.

Steve Gibson: Leo, great to be with you again. And we had a great time on Saturday doing our...

Leo: Man, you had this audience just rapt, watching every move. One guy told me, though, "I lost him at prime numbers." But for most - so Steve did a great SQRL presentation, like really detailed SQRL presentation. Not just how it works, but the genesis of the idea, the deep guts of how it works, really explained it to everybody. And we had a wonderful studio audience, about 45 people. All went over to Lagunitas afterwards. That was fun, too. Had a great time.

Steve: It was. And we had one listener flew in from, believe it or not, from Japan for this. And someone else from Dallas.

Leo: Another guy who was at the Boston event came in from Austin. Yeah.

Steve: Yeah, Austin, right.

Leo: So Boston and Austin. So, yeah, you have some hardcore fans. Took a lot of selfies again.

Steve: So we're helping people's frequent flyer miles in every way we can.

Leo: I love it. Anyway, it was a lot of fun. If you haven't seen it, it's the first episode on our new Events feed at TWiT.tv/events.

Steve: Yeah, I loved that. I didn't understand why I got number one. But it's like...

Leo: You're number one.

Steve: Event number one, yay!

Leo: Number two, three, four five, all the way up 100 will be CES. And then that's where we're going to put stuff that is out of the ordinary that we do, TWiT events. And then the news feeds will be the breaking news things we do like the Apple, Google, and Microsoft announcements, stuff like that.

Steve: Right.

Leo: So what's on the agenda this week?

Steve: So this is interesting. The title of today's podcast, #743, is Android StrandHogg, with two G's, H-O-G-G, which is a Norwegian word referring to mistreatment by Vikings. And one may wonder how that became our show title. It's because of something really interesting. And it's going to be interesting to see how this evolves. Four years ago, and we talked about this at the time, there was some research out of Penn State and FireEye, the guys at FireEye Security, noticing that there were some features in Android's multitasking system by design which they demonstrated could be abused. And Google said, okay, yeah, we don't really think that's a problem.

Well, it's now been found in the wild. And it's not good. So anyway, we're going to wrap up by talking about StrandHogg, which is the name that was given to the guys who first found it when banks were reporting that their customers' accounts were being drained as a consequence of what Google deliberately put into Android.

Leo: Wow.

Steve: So makes for an interesting story. But as we approach the end of patching for Windows 7, I wanted to revisit the free upgrade from 7 or 8 to 10, which it turns out can still be done. Also we have a very important alert for users of HP's SSDs because they're all going to die. We also have complications that arise with international privacy treaties, which I found sort of interesting, when end-to-end encryption might be threatened by one of the nations in the treaty. And what really brought me to this is that apparently they're still rattling their sabers at the DoJ about this and whether they're going to legislate us out of encryption, which, you know, 2020 may be really interesting.

We've also got the U.S. government's formal permission to hack them, that is, within the executive branch, believe it or not. We'll see how that goes over. We also have a quick look at a particularly devastating ransomware attack. I'm trying to downplay ransomware, but it's still there; and this one was just like, yikes, ouch. We also have more anti-tracking privacy happiness coming soon by default to Firefox. The never-ending headaches caused by Windows DLLs. An update, I wanted to check back in to see how my "Joy of Sync" recommendations and self-determinations have been fitting.

Leo: Oh, good. I've been waiting for an update. That's great.

Steve: Yeah. And then we're going to wrap up by talking about some, well, some questionable multitasking features that Google deliberately has in Android that are now looking as though they're not quite so well advised. So I think another great podcast for our listeners.

Leo: Uh-oh. Yeah. You know what, you're getting good at the teases because now I just have to listen. I just have to find out.

Steve: Yup. So our Picture of the Week is one I've been holding onto for a while showing a snippet of comment code.

Leo: I love this. I love this.

Steve: At the top of some code. And we've often talked about how some comments you're writing for yourself, some comments you're writing for whoever it is that gets stuck with the task of maintaining the code that you wrote. And of course ideally people who come along later will augment the comments that they find with more to keep them current. Anyway, so this is a comment block that reads: "Dear programmer: When I wrote this code, only god and I knew how it worked. Now only god knows." It says: "Therefore, if you are trying to optimize this routine, and it fails (most surely), please increase this counter as a warning for the next person."

Leo: Oh, my god.

Steve: And then the next line shows as a variable "total_hours_wasted_here =," and it's currently set to 254. So anyway, I just thought that was a hoot. So, yes. If you

encounter this, and the count is large, maybe you should consider not bothering to optimize...

Leo: Do we know where this came from? This is awesome.

Steve: No, I don't. Isn't it great? I love it.

Leo: Oh, man. I love it.

Steve: So, okay. It turns out today, even though Windows 10 free upgrade ended on the 29th of July 2016, you can still upgrade Windows 10 for free. And so I just wanted to mention that because I imagine maybe this month, maybe next month, because after all, as we know, middle of next month, middle of January 2020 is the last security update which Microsoft will be providing for Windows 7.

Now, we say that assuming that they're going stick to, and it seems likely that they will, stick to their determination to force anybody who wants more updates into the paid plan for the following three years. At the same time, we've seen them reach back even to Windows XP if something really horrific like BlueKeep, in this example, had happened. Because of course they did go back and patch even Windows XP because the BlueKeep vulnerability in the desktop server was so bad that they wanted to go back and fix it.

So presumably updates stop after the middle of next month, January 2020. And so I don't recall exactly what it was, or I did know as I was starting to do this research, but I remembered that there was some kind of, I don't know, skeezy way that you could still get an update that didn't really seem copacetic that Microsoft was making available. And I didn't know, if that was still available, what was going on.

And of course it never really was clear to me why this was time limited in the first place, why they were, after all this massive push - remember there was the, well, of course my own freeware, Never10. And then there was also the GWX, Get Windows 10, that was unwanted software that was being downloaded that was really pushing people. And for a while you could push it off, and then it sort of stopped giving you a choice. You had to hit the X button to close the dialog because you then were only left with a choice of upgrade now or upgrade tonight. And it's like, wait, what happened to "No, thanks"?

So after all that, then they suddenly said, okay, we're going to give you a deadline. And I guess the point was to give people the impression that this was really it. And then if you didn't get it now, then you're never going to be able to get it for free. And in fact, if you want Windows 10 Pro, oddly enough, the Download Windows 10 takes you to this page where, first of all, they're trying to sell you their Surface hardware because you fill out a questionnaire about whether Windows is fast or slow, whether you run one thing or more than one thing at a time. It sort of does this weird profiling of you. Basically the answer always comes up, oh, you should upgrade to a Windows Surface thing. But if you answer the questions in some way, then it will also give you the choice of, well, okay, we guess if you just upgrade to Windows 10 you'll be happy enough. So it turns out that...

Leo: By the way, I don't know what page you're going to. But I don't see that on this page.

Steve: Yeah. If you go down...

Leo: Windows 10 confirm.

Steve: No, I think if you scroll down further because there was something about having to get the license, to qualify for a license or purchase the license.

Leo: You don't even really need to do that because you don't need a license, is the point.

Steve: Correct. Correct. And that ends up being the case.

Leo: So I didn't have to do any of that. I'm just going to download right here the ISO, 64-bit. The key is in the past we've always thought, well, if you do that, you're going to have to enter a serial number.

Steve: Correct. And they say you're going to.

Leo: And they say you're going to. But you don't.

Steve: Yes. And are you running this from Windows 7?

Leo: No, this is on a Mac.

Steve: Oh, okay. So maybe it's different if you're running it from Windows 7.

Leo: I've not seen - there may be other entry points. I just googled "Windows Media Creator Tool" and went right to that page. There may be other entry points that you've gone through.

Steve: Well, so I created a shortcut for our listeners, grc.sc/win10. So that takes you to - that's just a shortcut to the official Windows 10 software download, which does eventually take you to the Media Creators Tool. That's where it gets to.

Leo: Well, yeah. So don't follow your link. Just google "Media Creation Tool." It takes me right there. This is the number one link, and I didn't have any of that crap. It's a simple click. So I think you're maybe going in through a sales portal of some kind.

Steve: Create Windows 10 installation media.

Leo: Maybe you're right. Maybe if you're on Windows 7 it senses. It has a sense. But in any event, you don't need to have a key. It'll download it for you. Their theory is,

well, you're going to download a trial version. Eventually you'll need a key. You'll have to activate it in, I think, 90 days. But apparently not.

Steve: Yes. And in fact Ed Bott, who we know over at ZDNet, he was curious about all this. He wrote that he had decommissioned a machine back in 2017, a little Intel small form factor PC. And he was curious about the whole upgrade process. And so he went through this process and was told that he was going to have to have, like, purchase a license to Windows 10. But to his surprise, as he wrote, once he went through this upgrade process, he was greeted with a screen that I have in the show notes just saying "Windows 10 Pro. Windows is activated with a digital license."

You do have to go to the activation service. But it just moves you along. So you and I have both told our listeners what I wanted to, which is, even today, even though they sort of say you need to purchase a license - and last night when I was doing it from a Windows 7 machine, I was following through the Download Windows 10, and it took me through this purchase process.

Leo: Yeah. I clicked your link. And for me on this Mac it did the same, the proper page without any upsell. So I think it's because you were on Windows 7.

Steve: That would make sense.

Leo: Now, there should be a caveat issued here because nobody - Ed Bott, Paul Thurrott - nobody has been able to verify that this is Microsoft policy. What's not clear - and I just want to say it's not guaranteed to work because Microsoft says it doesn't work.

Steve: Right.

Leo: It seems to work with everybody we talk to.

Steve: Right.

Leo: But here's the reason I say that. If it doesn't work, you're now going to have installed Windows 10 on top of your Windows 7, and that may be a pain in the butt for you. There is a rollback; right?

Steve: Yes, yes. You are able to...

Leo: Okay. So you could theoretically roll back.

Steve: Yes, you are able to back out.

Leo: I'd still back it up before you do this would be my advice.

Steve: And I did also pursue the state of the online dialogue about this. And the only problem that anyone reported was the typical, like the upgrade hung because of some random hard drive or the USB, I mean, like there are various reasons why Windows 10 upgrade just fails. And so when it succeeds...

Leo: Microsoft hasn't fixed that part yet.

Steve: No. So when it succeeds, you seem to be golden. But as a consequence of the fact that it sometimes fails, all of that rollback stuff is there.

Leo: And you'll be glad if you have an image, too. So just for safety's sake.

Steve: Yeah. That would be a good thing to do. Make an image. But I just wanted our listeners to know, if they're feeling like, oh, boy, I sure do wish I had gotten it when I could, well, I don't think it'll ever not be available.

Leo: I did see one post on Reddit from somebody who says he worked in the Windows division, and that this is always - you probably saw it, too. This has always been the policy. They don't announce it because they want to make money if they can because it is a $130 upgrade if you pay for it. But he said it comes from Terry Myerson, who's been gone for about a year from Windows, who really was trying to get the upgrade numbers, the percentage of upgrades high, didn't really care. Revenue, Windows revenue is no longer that important to Microsoft. Well, I don't - it's not fully credible that Microsoft would turn its back on $130 from every upgrader. They may be a little bit more nervous about having a lot of Windows 7s out in the wild. I would be.

Steve: Well, yeah. And remember we covered it at the time. They were pushing to their shareholders or their stockholders like all of the, what was it, the monetization opportunities which would be made possible by all the things in Windows 10 that all of us dislike. So it's like, okay, well, that's all still there. So, yeah.

Leo: The other thing that's a question mark is whether you need to install on top of an existing Windows 7, or you can do a clean install. And most of the people I've seen say best to just install on top of.

Steve: Yes. I would also do that because it might not know. I mean, so you have the option. You will be presented with the option of creating installation media or just upgrading this system that you're running on. And for the sake of preserving and essentially promoting your Win7 or Win8 license to Win10, I would say just upgrade over. But yes, make an image first.

Leo: Some people have done clean installs and it's worked.

Steve: Oh, good.

Leo: The other thing I would say that's important for everybody to know is the way activation now works, once Windows 10 is activated on a machine, at that point you can wipe it, you can do anything you want. You have what Microsoft calls an "entitlement" to Windows 10. That machine for thereafter, as long as you don't change major system components, will be activated for Windows 10. So a clean install after the fact is fine.

Steve: Yeah. And you and I have always been believers of clean install. I don't think I have, in fact I am sure I have never once in my life done a major Windows version upgrade. It's like, it's not worth it. Just start over.

Leo: If you have a version of 7 installed, go ahead and install on top of it. If you don't, go ahead and try because people in the chatroom are saying, no, no, it works with a clean install, too. So this is what's so strange. Microsoft has been dead silent on all this.

Steve: Yes.

Leo: And who knows? After they hear this show, they may flip off the switch; right?

Steve: Whoops, we forgot about that page, yeah.

Leo: But for now it works. Now, here's the real question, Steve. Should you upgrade to Windows 10?

Steve: Hmmm. That is a question. I mean, I'm liking...

Leo: Would you?

Steve: Not immediately, as we know. I mean, I'd still be on XP if my XP machine hadn't just finally fallen off a cliff. It just decided, okay, I'm tired.

Leo: But you're a security expert. I wouldn't recommend most people run XP online; right?

Steve: No, no, no. No, no. And I absolutely agree. I've made peace with Win10. I have it on a bunch of systems. I'm comfortable with it. I have access, thanks to my MSDN subscription, to the long-term servicing channel, so I'm able to install a machine that doesn't have any of the cruft on it. And I have decruftified a number of standard Win10 Pro machines to get the Candy Crush Soda Saga and all that other nonsense off of it.

Leo: There's a PowerShell one-liner that will delete all of those apps.

Steve: Yes.

Leo: That's what I always do. I mean, complaints some people have had, I know you've had, is that you can't really fully turn off the telemetry, the phoning home that Windows does. But I don't know. If you're going to use Windows, you might as well live with that.

Steve: Yeah. And of course we know that there are some small percentage of people who have real problems with it. But again, it's a small percentage. They're loud. And Paul and Mary Jo cover those sorts of things when they happen.

Leo: Oh, yeah. We hear from them.

Steve: And unfortunately it does create reputation damage for Windows 10 because people hear that, and they go, I don't want any of that.

Leo: I honestly, personally, I think it's as good as Windows 7, which I think was the best version of Windows Microsoft ever made. I think 10 is fine. I'm not a Windows fan. But 10 is no worse than any other Windows, and it's better than most. How about that?

Steve: No. Yeah, I agree. And the only thing I wish is that this whole rolling Windows forward thing, which continually creates instability, I wish they'd just let it alone. And I heard Mary Jo saying that last week.

Leo: Oh, yeah.

Steve: Just leave it alone for a year instead of continuing to fuss with it.

Leo: They kind of did this. The most recent update, 1909, was barely a feature update. It was really more just a cumulative update of bugs. And I think that that - everybody loved that. Everybody said, oh, thank you. To me...

Steve: Did you see that they've announced now formally what the next one is going to be?

Leo: 2004, yeah.

Steve: I said to Lorrie, yeah, I said to Lorrie last night, now, this is not going to confuse anybody.

Leo: No.

Steve: Because we're going to have Windows 10 2004. It's like wait, what?

Leo: Huh?

Steve: In 2020 we're going to get the 2004 version of Windows 10? What?

Leo: Yeah, Paul and Mary Jo don't like it either, yeah.

Steve: What are they thinking? Okay. Well, speaking of numbers, we are now counting down to 32768, which the programmers among us know is 15 bits in binary. 65536 is the full 16 bits. So half that is 32768. Unfortunately, due to a mistake in the firmware running a large set of Hewlett Packard Enterprise Class SSDs, the instant the total power-on running time of any of those SSDs crosses 32,768 hours, 32768, which is three years, 270 days, and eight hours, all of those drives will simultaneously become totally and unrecoverably offline, taking all of their stored data with it.

Leo: Oh, my god. That's terrible.

Steve: It's a catastrophe. So HP's customer bulletin, I have a link in the show notes, says: "Bulletin: HPE SAS Solid State Drives - Critical Firmware Upgrade Required for Certain HPE SAS Solid State Drive Models to Prevent Drive Failure at 32,768 Hours of Operation."

Leo: That's a really concrete NTBF. They all fail at 32768. All of them.

Steve: Yes. In fact, it even suggests that, if the drives were simultaneously commissioned into a fault-tolerant RAID, they would probably all fail at the same time.

Leo: Now, these are enterprise drives. So I'm going to guess they're probably not in the kinds of PCs our listeners are buying, unless they...

Steve: Well, we've got, as we know, we have a lot of high-end listeners.

Leo: Oh, yeah.

Steve: Yeah.

Leo: But not like in your HP laptop. That's even a different company these days.

Steve: Right, right. So they said: "IMPORTANT: This HPD8 firmware" - that's what you want to get, you want the HPD8 firmware - "is considered a critical fix and is required to address the issue detailed below. HPE (HP Enterprise) strongly recommends immediate application..."

Leo: Yeah. Yeah.

Steve: Yeah, that'd be handy - "of this critical fix. Neglecting to update to SSD Firmware Version HPD8 will result in drive failure and data loss at 32,768 hours of operation and require restoration of data from backup in non-fault tolerance, such as RAID 0 and in fault tolerance RAID mode if more drives fail than what is supported by the fault tolerance RAID mode logical drive. By disregarding this notification and not performing the recommended resolution" - get this - "the customer accepts the risk of incurring future related errors." They said: "HP was notified by a Solid State Drive manufacturer of a firmware defect affecting certain SAS SSD models..."

Leo: Oh, so it wasn't HP software. Their supplier.

Steve: Yeah, exactly, "...used in a number of HP server and storage products." And now here they are: the HPE ProLiant; the Synergy; the Apollo; the D3000, D6000, and D6020 disk enclosures; MSA Storage; StoreEasy 1000 Storage; StoreVirtual 4335 Hybrid Storage and StoreVirtual 3000 Storage are affected. Then they said: "The following platforms are not affected by this issue: the HPE 3PAR StoreServ Storage, the D8000 Disk Enclosure, the Nimble Storage, Primera Storage, StoreOnce Systems, XP Storage, and SimpliVity," which is hard to say.

Leo: That sounds like a Simpsons name.

Steve: SimpliVity. SimpliVity. SimpliVity. There it is, SimpliVity, yes, so well named, SimpliVity. And they finally said: "The issue affects SSDs with an HP firmware prior to HPD8 that results in SSD failure" at that time. "After the SSD failure occurs, neither the SSD nor the data can be recovered." In other words, it bricks itself. It hard bricks itself at 32K hours. "In addition, SSDs which were put into service at the same time will likely fail nearly simultaneously." Yikes.

So in the disclosure they list the 20 SSD model numbers. So anyone who's listening who worries they may be affected, I would take this seriously. And they do have firmware update software for Linux, Windows, VMware ESXi, provided from links at that. So wow. And it's not clear that everybody who has these is on the mailing list and is going to see this announcement. So I do hope that this information gets picked up and covered enough that people aren't going to be hurt.

Leo: I'll say it on all our shows through the rest of the week because there's people need to hear it, yeah.

Steve: Yeah, because it's belly-up.

Leo: No matter what we do, though. There's going to be people lose data. It's going to happen.

Steve: Yeah, yeah. For example, as we know, how many BlueKeep vulnerable RDP servers are currently exposed? Well, we know it's about half a million, despite the fact that the news couldn't have been any more well covered than it has been.

Leo: Can you believe there are people who don't listen to Security Now!?

Steve: Doesn't matter.

Leo: What is wrong with them?

Steve: Well, apparently fewer every day, Leo. So that's good.

Leo: We're going to take over the world at this rate in the year 32768.

Steve: Yeah, wow. So it turns out the EU is not happy about a possible U.S. encryption ban. And when I read that headline, I thought, wait, wait, what possible U.S. encryption ban? Well, okay. So it's nothing that our listeners haven't heard about. Although it turns out that the ongoing battle over end-to-end encryption took a bit of turn last week when EU officials warned that they may not take kindly to a U.S. encryption ban or the insertion of a crypto backdoor in the U.S.-based technologies. And again, as I said, 2020, it really does promise to be an interesting year.

Back in June of this year, it turns out that there was a little-publicized meeting where senior U.S. government officials quietly met to discuss whether they could legislate tech companies into not using unbreakable encryption.

Leo: Whoa. I don't remember that.

Steve: Unh-unh. It was not covered. There was apparently a story in Politico that covered the NSC, the National Security Council...

Leo: Oh, I do remember that. But I just dismissed it as political posturing.

Steve: Right.

Leo: I do remember this story, yeah.

Steve: Right. Because they were wondering whether to ask Congress to outlaw - outlaw - end-to-end encryption. So of course let's just punt this one. So U.S. officials did not reach a decision one way or the other on the issue. But news of the conversation spooked some enough to ask the EU some formal questions which were picked up by somebody at Techdirt, Glyn Moody at Techdirt. The person asking the EU was someone named Koerner, who asked whether the Commission would consider a similar ban on encryption in the EU. He also asked what a U.S. ban on would mean for existing data exchange agreements between the EU and the U.S.

And it turns out that that actually does affect our various treaties. So the question was posed, would a ban on encryption in the USA render data transfers to the U.S. illegal in light of the requirement of the EU GDPR for built-in data protection? So at the moment, our two regions, the EU and the U.S., enjoy an agreement known as EU-U.S. Privacy

Shield, which they introduced after the European Court of Justice invalidated a previous agreement which was called the International Safe Harbor Privacy Principles.

So today's Privacy Shield, which is what we have now, is a voluntary certification scheme for U.S. businesses. By certifying under the scheme, U.S. companies prove their adequacy to transfer and process data on EU citizens. It shows that the companies have made some effort to follow Europe's strict privacy principles in the absence of any cohesive federal privacy law in the U.S. So it's sort of a stopgap, let's keep doing things while we figure out what direction the U.S. is going to take.

On November 20th, just a couple weeks ago, European Commission officials replied with their answers, confirming that they would not consider a ban on encryption in the region and pointing out that the General Data Protection Regulation (GDPR) explicitly refers to encryption as a "privacy protection measure." But the answer to the next question was a bit more contentious. They said: "If the U.S. were to enact new legislation in this area, the Commission would carefully assess its impact on the adequacy finding for the EU-U.S. Privacy Shield" - which as we just said is that framework which the Commission has found to provide a level of data protection that's essentially equivalent to the level of protection in the EU, thus allowing for the transfer of personal data from the EU to participating companies in the U.S. without any further restrictions.

So the jury's still out on how the EU would react to cross-Atlantic data transfers if the U.S. were to implement crypto backdoors, which it seems unlikely that end-to-end encryption itself would be outlawed. But we keep having U.S. law enforcement being unhappy about the going dark problem and their inability to serve a warrant, at least, which would compel the decryption of specific conversations.

So there's an attorney, Ashley Winton, who is a U.K.-based specialist in privacy law, who explained that a split between the two territories on data exchange could have serious consequences. He said: "We know that under the GDPR personal data must be held securely; so legislating against strong encryption or introducing legal backdoors is not going to be good for the safe passage of European Personal Data, howsoever it gets to the U.S. Unlike the annual review of Privacy Shield, in theory, if the European Court were to determine that the transfer of Personal Data to the U.S. was no longer safe, all affected transfers could be halted immediately." Well, that's not going to happen. But apparently it's going to cause some sort of kerfuffle.

So anyway, I just thought this was interesting, that we're sort of sitting here worrying about what the U.S. is going to do and what's going to happen with our technology. It hadn't really occurred to me that Europe is looking at the strength of their GDPR and essentially requiring that, to the degree that U.S. companies end up with EU citizen data, our U.S. companies need to be offering a similar level of protection. And it may well be that law enforcement requiring the ability, depending upon the way it's implemented, I mean, again, "backdoor," as we know, is a heavily freighted term. But it may cause some ruffling of treaties between the U.S. and the rest of Europe. So it'll be interesting to see how that develops.

And Leo, let's take our second break. And then we're going to talk about this interesting thing that happened in Washington, where agencies of the government are being encouraged to allow themselves to be hacked.

Leo: Okay. Yeah, I remember when the NSC considered it. And I think we may have reported on it. But given that by the time the story came out in Politico, they'd already declined to do it. And given the composition of the NSC at the time, I didn't judge it to be a serious interest in doing this. And you could see why. Somebody might have said, what if we did this? And then somebody maybe more astute said, if

we do that, we can't do business with the EU anymore. Might have just killed it; right? I mean, I don't know how strong the interest is or was in doing it. I hope it wasn't strong. And we know that the law enforcement wants to do it.

Steve: Well, but Leo, we know it hasn't gone away.

Leo: No. The FBI director asked for it, and a number of people have asked for it. Let's just hope saner heads prevail.

Steve: As people say, you may ask.

Leo: You can ask. I think also the fact that the EU said this might have been also a little bit of a, hey, guys, over here. If you do that, just be ready because it's going to cause some waves. That may be the whole point of the EU saying this; right?

Steve: Right, right, right.

Leo: Just a little reminder that it affects everything.

Steve: It's not all about you guys, yes. It's not all about you Yanks.

Leo: Yeah. I think that's probably what's going on there. Just in case anybody thought this was a good idea. Back to Steverino.

Steve: So under the heading of this should be interesting, the Cybersecurity and Infrastructure Agency (CISA), which we run across that acronym from time to time, they're active and doing things. It's part of the Department of Homeland Security. They've just published what they call the VDP, the Vulnerability Directive Policy, requiring executive branch federal agencies to be welcoming and responsive to cybersecurity bug reports from the general public.

I have a link in the show notes to the CISA.gov page. And they said: "A VDP directive and you." They said: "Today we're issuing a draft binding operational directive, BOD 20-01, which will require federal civilian executive branch agencies to publish a Vulnerability Disclosure Policy. A VDP allows people who have 'seen something' to 'say something' to those who can fix it. It makes clear that an agency welcomes and authorizes good faith security research on specific Internet-accessible systems.

"In preparing this directive, we worked with several agencies that have VDPs and made an effort to align the directive with federal guidance, international standards, and good practices. But this directive is slightly different from others we've issued, where agencies are directed to take an action, and then CISA verifies the action has taken place. Here, while agencies must maintain VDPs that are the beneficiaries of vulnerability reports, it's the public that will provide those reports and will be the true beneficiaries of vulnerability remediation. That's why we're doing something we've never done before with our directives: seeking public feedback before issuance.

"We want to hear from people with personal or institutional expertise in vulnerability disclosure. We also want to hear from organizations that have a VDP and manage coordinated vulnerability disclosures. In seeking public comment, we're also nodding to the fact that, to our knowledge, a requirement for individual enterprises to maintain a vulnerability disclosure policy has never been done before, and certainly not on this scale.

"So what does the draft directive do? It lights a fire. Each agency must publish a VDP and maintain handling procedures, and the directive outlines a set of required elements for both. It draws a line in the sand. Systems 'born' after publication of a VDP must be included in scope of an agency's VDP. It expands the circle. Until everything is included, at least one new system or service must be added every 90 days to the scope of an agency's VDP. It starts the clock. There's an upper bound - two years from issuance, in this draft - for when all Internet-accessible systems must be within scope. All are welcome. Anyone that finds a problem must be able to report it to an agency. No 'catch and keep.' An agency may only request a reasonably time-limited restriction against outside disclosure to comply with their VDP. And defense, not offense. Submissions are for defensive purposes. They don't go to the Vulnerability Equities Process."

And then they had two points for what it doesn't do: "It does not establish a 'federal bug bounty.' A bug bounty is a program that pays researchers" - as we know - "for valid and impactful findings. Nothing in this directive prevents individual agencies from establishing a bug bounty of their own, though. And does not create a 'national VDP.' The directive is an executive branch policy instruction that requires federal civilian executive branch agencies to have a VDP. The difference might appear slight, but they're very different things."

And then it says: "Why isn't this a national VDP? We think a single universal vulnerability disclosure policy for the executive branch is a good goal. It makes sense particularly when each agency has all Internet-accessible systems in scope, but we expect that goal to be an unrealistic starting place for most agencies. Instead, the directive supports a phase-based approach to widening scope over time, allowing each enterprise - comprised of the humans and their organizational tools, norms, and culture - to level up incrementally."

Then they finish with: "Doing good things together. We believe that if you make good things easier to do, more people will do them. With this directive, we want to take steps that diminish complexity and make expectations plain. In support of that, we're also sharing draft implementation guidance on the directive, as well as a draft VDP template." They said: "We welcome your feedback and perspective on all these documents, as well as any comments on our approach. Public comment will take place on GitHub and last until December 27th," so basically for one month. I've got links to the details in the show notes, and I think this represents a welcome and really forward-looking and insightful step forward for the U.S. government.

I'm sometimes surprised, but I'm really pleased to see it, basically encouraging, explicitly encouraging the publication of formal guidance for people who find problems with government systems. You're not risking, probably, given that you provide the information to the relative agency in a responsible fashion, you're not going to have the feds knocking on your door, hauling you away to jail, given that you've disclosed responsibly. And the idea that it's going to be requiring, every 90 days, something else needs to be put in scope; and, regardless, everything must be within scope in two years. It means that two years from now everything that this VDP policy covers will be subject to scrutiny from the outside world, and the nature of responsible disclosure explicitly laid out. So bravo, surprisingly so.

Leo: I also want to say bravo to Jeanette Manfra, who - I don't know who that is. She wrote the memo. It is the best written bureaucratic memo, I mean, it has "pineapple pizza" in it. She is a good writer. And maybe that's...

Steve: I agree with you. I had this - yup.

Leo: Yeah. She's assistant - she's not some flack hired to write these, either. She's the Assistant Director for Cybersecurity for the Department of Homeland Security. So she is a fairly high up bureaucrat. But man, can she write.

Steve: I agree. I thought it was really well written.

Leo: Oh, she's a former communications specialist and military intelligence office. Wow. They're doing something right there, I'll tell you.

Steve: Yeah, nice.

Leo: Have to get her on one of our shows. Do you think she's be able to talk about stuff? This is a really interesting...

Steve: Yeah. Oh, like without being bound by crippling non-disclosures.

Leo: Yeah. She says: "At CISA we work to do good things. Some are easy, like eating pineapple on pizza." And she has a link, by the way. "Some are hard, like managing risks in 5G. Yet we know if it's hard to do good things, most people won't do them. And reporting a vulnerability on a government system shouldn't be so hard."

Steve: Yes.

Leo: It's really well written. Anyway, sorry, I just - I threw that in because I...

Steve: No, I'm glad you did because that helps...

Leo: You don't usually see that.

Steve: Yes.

Leo: By the way, the pineapple pizza takes you to an article on understanding foreign interference, "The War on Pineapple."

Steve: Okay.

Leo: This is interesting. Targeting divisive issues, like pineapple on pizza. Moving accounts into place. This is actually a really good slide show on how foreign interference works. "Pineapple Pizza Controversy Rocks the U.S."

Steve: Oh, right, fake news.

Leo: Fake news.

Steve: Yup.

Leo: Wow. I'm impressed. Wow. I'm sorry. Didn't mean to derail you.

Steve: No, no. I'm glad. I'm glad for that.

Leo: She deserves some credit.

Steve: So as our listeners know, last summer was the summer, sort of before the school year, the summer where I had a hard time, I think ransomware was the overwhelming topic of the podcast for three or four weeks running because there was just so much of it happening. And I promised I wouldn't do that anymore. But sometimes something that's particularly egregious happens, and so it's necessary to just take a moment and mention it.

In this instance we have - this is like the big win, unfortunately, for ransomware guys, is when, and we've talked about this also, the managed service provider, the idea that there is a single organization that is providing managed services to some large network of their customers and clients. And something happens at the managed service provider to let bad guys get in. They then set up shop; they figure out like where they are; and they say, "Holy crap, look what we have access to." And then they of course are able to essentially create a devastating attack.

In this case, this is a healthcare managed service provider that was responsible for providing services for 110 nursing homes. So think about that, 110. And also some acute care facilities. All of them have been simultaneously crippled by a ransomware attack on their common IT provider whose name is Virtual Care Provider, Inc. (VCPI). They're a Wisconsin-based managed care provider. They provide data management and records hosting, security and access management to nursing homes across the U.S.

Brian Krebs picked up on this last Monday while the attack was still underway. He said that it involves our old friend Ryuk. And as we know, the hackers who were driving Ryuk are known to calculate how much ransom they feel the victimized organizations can be forced to pay based on the size and perceived value of the destruction that they have wrought.

But we've also seen instances where the crooks get it wrong. And that appears to be the case here. VCPI's chief executive and owner, Karen Christianson, told Brian that her company simply cannot afford to pay the $14 million bitcoin ransom. I mean, you know, this is - yes, they're a managed care provider for 114 healthcare facilities. But it's a competitive industry. They don't have $14 million. And apparently no insurance. She said

they can't begin to pay the $14 million ransom. As it is, it turns out, VCPI's own employees have been asking when they'll get paid because it's their own salary infrastructure was also hit. But she said the top priority is to wrestle back access to the lost electronic medical records.

The attack successfully affected all of the firm's core offerings - Internet service, email, access to patient records, client billing, and phone systems - and even the internal payroll operations that VCPI uses to pay its own workforce of nearly 150 employees. Regaining access to electronic health records is the top priority because, without that access, the lives of the seniors and others who reside in critical-care facilities are at stake.

Christianson said to Brian: "We have some facilities where the nurses can't get the drugs updated and the order put in so the drugs can arrive on time. In another example we have an assisted living facility that is just a single unit that connects to billing. If they don't get their billing in to Medicaid by December 5th" - that's two days from now - "they close their doors. It's over, and seniors who have no family to go to are then suddenly homeless," Karen said.

Leo: Oh, my god. These people, these ransomware guys are jerks.

Steve: I know. They said: "We have many clients right now who are demanding, 'Just give us our data,'" she said, "but we can't. We don't have it." So imagine how devastated she must be.

A report from Vanderbilt University's Owen Graduate School of Management noted that the corrective actions being taken to harden and secure facilities and systems can, at least temporarily, worsen the problem. They wrote: "Corrective actions are intended to remedy the deficiencies in privacy and security of protected health information. However, enhanced security measures may introduce usability" - which they said "we define as the ease of use" - "problems. New security procedures typically alter how clinicians access and use clinical information in health information systems and may disrupt the provision of care as providers require additional time to learn and use the new and modified systems." In other words, yeah. Additional layers of security can be annoying. But of course they can prevent this.

But anyway, obviously, the group behind Ryuk are serious. And in another similar attack there were hundreds of veterinary hospitals were similarly affected. So I just wanted to point out that these sorts of cyber attacks that are of this nature, I mean, are having devastating effect in the real world. And as you said, Leo, these guys are just real creeps.

Leo: They're also stupid because they're asking for more than the company can pay. They're never going to get money. So it probably is a 14 year old, I mean, with no real life experience. I mean, that's too bad. That's the problem with Ryuk is a script kiddie can do it now. Anybody can do it.

Steve: Yeah, exactly. And Brian reported that the attack was unleashed inside VCPI's networks around 1:00 p.m. Central time on the 17th of November. So it's been happening for some time. It could have been lying in wait for a while as the intruders mapped out the internal networks and compromised the resources and data backup systems in preparation for the ultimate attack. Because of course they want to nuke the backups also, if they have any opportunity to do that, because that renders the victims

utterly helpless. And so apparently these people were acting as responsibly as they could, but the bad guys were in there long enough to be able to find and compromise their backup systems, as well. So we just can't say, well, yeah, you deserved it because you weren't doing the right thing.

And this takes me back to the point I've been making about any of our listeners who may be in enterprises using managed services. What tends to happen is that the managed service provider just says, well, give us Remote Desktop access into your network. We talked about last week how most managed service providers have admin-level access into the networks of the clients that they're providing services to. Well, yeah. That means if something gets into the managed service provider, they have admin access into the networks of their clients. You don't want that to be you. Generally, that level of access isn't needed.

So everyone wants it because it's easy, in the same way that we used to all just want to run as admin because, oh, look, I can do anything I want to without having to switch users. And we've learned the lesson of how that's not safe. So, boy, be really careful. It's not just your own network that you need to be careful about. You really need, essentially, as IT, you're indirectly taking responsibility for the security of anybody who you let in. And managed service providers are getting knocked over. So I just wanted to drive that point home again.

Firefox is seriously pushing back on tracking signal leakage. As we know, cookies are not only the most obvious, sanctioned, and most easily controllable aspect of web browser-based Internet tracking. They are what people use by default. But as we've talked about, "fingerprinting" makes use of many subtle signals the browsers may send in an attempt to lock onto a user who is deliberately attempting to thwart cookie-based tracking.

And as it turned out, when I was putting this together last night, I realized, oh, yeah, me, too. I was sitting in front of a very wide screen. It's 3840x1600 screen. And so it's impractically, I think it's probably the one that Lisa has, Leo, because I know you have mentioned, I've heard you say that Lisa has also a really wide...

Leo: Oh, it's so amazing. Every time I see it, I say, "I want this."

Steve: And so you don't run that. You don't run a browser full screen on a screen that wide. It doesn't need it. It's ridiculous.

Leo: Right. No, no.

Steve: So I realized that I had the browser that I was using as I was pulling the show together last evening, it was on some random arbitrary slice of the screen, meaning that its x-coordinate and its horizontal width were random, well, they weren't random numbers, but they were just some, you know, they were not standard numbers. It wasn't X of 0 and width of 1600, which is like a standard resolution. I was at some particular 1326 X and 1842 in width. Which is something that script running in my browser, that is, script running in an ad is able to see. Which means, so long as I don't move my window around, and generally when I relaunch my browser it comes back where I last used it. So that tends to be kind of a generally sticky little bit of beacon information that I'm sending out as I move around the web.

And there are, as we know, all kinds of similar soft signals which, when they're merged together, can serve to profile us. Things like what time zone we're in, the exact pixel

layout of our browser, which fonts we have installed in our system. And so just deleting cookies no longer tells the whole story. We talked, in fact, about how there was a navigator.getBattery function which was allowing advertising scripts to determine the exact battery charge of the machines their users were using, and note that it tends to change predictably over time. I mean, we tend to think, oh, my lord, nobody is going to go to the trouble of, like, monitoring that. It turns out people are.

So anyway, that's the backdrop against which Firefox has been very vocal about introducing explicit anti-fingerprinting code into the base browser. It's there now, but it is not turned on by default. And we've talked about various of these things over time, like it's even possible for an ad to draw something in, like draw something with vector art and then snap the pixels that result, and look at subtle variations in the way lines are drawn and rendered into digital pixels.

Turns out those things vary subtly from OS to OS and from version, even version of operating system, and browser to browser. So there's that. Absolute screen coordinates would be obscured. The image extraction would be blocked. They're considering rounding window dimensions to multiples of 200x100. Only specific system fonts would be enumerated. The time precision would be reduced to 100ms, and 100ms of deliberate jitter would be added. The keyboard layout would be spoofed. The locale would be spoofed to simply English and U.S. The date input field and the date picker panels would be spoofed.

I mean, clearly the bad guys have gone to so much effort in order to pick up on signals. The time zone that we're in would not be accurate. It would be spoofed to UTC so that everything would be uniform. And all device sensors would be disabled. Turns out you can read things like position of the device that the user is holding. So of course the downside to all of this is that any websites that were making legitimate and beneficial use of those details, like we've talked about how could you use battery level, well, maybe if you were a particularly conscientious gaming app you might back off on the amount of high-energy GPU usage if you saw that a laptop's battery was getting lower, or you might proactively say to the user, hey, you'd better save your game, looks like things are getting dangerously close to low.

So there are useful things this can be done for, or useful applications for these things. But unfortunately it turns out the actual users of these, as instrumentation has shown, are the bad guys rather than the good guys. So despite the fact that all of these features have been added to our browsers over time, they're being used against us more than for us. So the good news is, as we know, there's a tyranny of the default. Most users are running Firefox just out of the box. Our listeners probably already have fingerprinting turned on. I know I do.

The good news is we're now at v70. Two major versions from now, at Firefox 72, if everything continues to go smoothly, that fingerprinting default will be switched to on. So as Firefox is updated across the board to v72, this kind of long-term fingerprint tracking will, at least for Firefox users, become increasingly difficult and, arguably, soft to a point that it no longer provides sufficient precision for the bad guys to even bother with it, which will be a nice thing.

And our last little bit of news for the week, as you noted on the Sunday show, Leo, it was a slow news week because the hackers in our case took Thanksgiving off. But there was one more piece of news that I thought was interesting, which is new problems with Windows DLLs, or I guess I should say "continuing problems with Windows DLLs." As we know, some ideas in computer science are fundamentally fraught with problems. An example would be the oh-so-convenient, though horrifically insecure, practice of allocating temporary communications buffers on a stack that is shared with code execution history and pointers. What could possibly go wrong?

Another historically bad idea has been Windows dynamically linked libraries. We know that Windows designers had their hearts in the right place. Back in the 1980s when Windows was attempting to run in a system that had a 10MB hard drive, actually I guess the early Windows ran off of floppies, and 512 kbytes of RAM, there was not a single byte to waste. So there was tremendous pressure to share any common functional code among the system's components and applications.

The idea of the DLL was clever in that not only could there only need to be one DLL stored on the hard drive; but, even more importantly, only one copy would ever be physically loaded into physical memory. When another program was run and wished to use some of the functions from one of the shareable DLLs, the image of an already loaded instance of that DLL would be mapped into the virtual memory space of the requesting app by the Windows loader so that even apps that needed the same things would not be causing a second copy to occupy RAM. This allowed much more to be accomplished with a small disk and a small RAM footprint.

But this technology did not age well. Over time, there was a divergence of DLL versions and capabilities. DLLs could use other DLLs. And that created a confusing network of inter-DLL dependencies. Such a mess arose that Microsoft was finally forced to step in and attempt to fix this problem.

Quoting from the Wikipedia entry on what's known as "side-by-side assembly," Wikipedia says: "Side-by-side assembly (SxS, or WinSxS on Microsoft Windows) technology is a standard for executable files in Windows 98 Second Edition" - all the way back then - "Windows 2000, and later versions of Windows that attempts to alleviate problems, collectively known as 'DLL Hell,' that arise from the use of dynamic link libraries in Microsoft Windows. Such problems," Wikipedia writes, "include version conflicts, missing DLLs, duplicate DLLs, and incorrect or missing registration. In side-by-side, Windows stores multiple versions of a DLL in the WinSxS subdirectory of the Windows directory, and loads them on demand. This reduces dependency problems for applications that include a side-by-side manifest."

They said: "Microsoft Visual C++ 2005 and 2008 employ side-by-side with all C runtime libraries. However, runtime libraries in Visual C++ 2010 no longer use this technology; instead, they include the version number of a DLL in its filename, which means that different versions of one DLL will technically be completely different DLLs now." I mean, this has just been a disaster.

So as I said, Microsoft designers were doing what they needed to at the time. And if anything, with the unfair benefit of 2020 hindsight, Microsoft might now be criticized for not completely killing off DLLs a long time ago. But one thing that Microsoft has always provided has been rigorous backward compatibility, pretty much at any cost. And, boy, trying to corral and manage what has happened with DLLs is the price that they are now paying.

So this brings us to today, or rather yesterday, when researchers with SafeBreach Labs published a trio of security advisories which described DLL-related bugs occurring in Autodesk, Trend Micro, and Kaspersky software. All of the problems were responsibly disclosed to their respective software publishers before their public disclosure, which was made yesterday.

And interestingly, all of these problems still relate to the original way that DLLs work. We've talked about some security problems in the past where, for example, if an app references a DLL, the Windows loader looks around the system for the DLL. Typically it looks first in the app's own directory, which thank goodness allows apps to bring a copy of the DLL with them. One of the many problems which DLLs have had has been newer apps would overwrite the DLLs with the same name, but a different version number, and

that DLL would act in some different fashion, which would cause - remember those days, Leo, where you would install some new software on Windows, and suddenly something that used to work before would break?

Leo: Oh, yeah.

Steve: It was - oh, my god.

Leo: The worst offender was the C++ libraries. And if you look in your Windows install even today, you probably have multiple C++ DLLs from a different variety of versions. And it can really screw with you. It is DLL Hell.

Steve: Yeah. It's been a complete collapse of that model.

Leo: The idea is great. These are dynamic linked libraries. The idea is the application doesn't have to have everything in the application. It can link to libraries. But somehow it just got out of control.

Steve: Just, yeah. Even Windows 98 Second Edition was already beginning to do some sequestering of DLLs because it was like, oh, ugh. Now we have lots of memory. We've got lots of drive space. We just wish everyone would bring their own code with them and not rely on anybody else because, oh, my god.

Leo: Or Microsoft should standardize on a single C++ runtime and deprecate all the rest. And if you try to install an older one, I don't know, seems like it's Microsoft's fault somewhat, but maybe not. Maybe not.

Steve: Yeah, no, I agree. It was a good idea. But really, you know, Apple has a reputation for killing things.

Leo: Yeah. This doesn't happen on Apples, yeah.

Steve: Right. But the flipside is Apple does get criticism for killing off things. Like they said, okay, we're no longer going to support such-and-such.

Leo: Also Apple, if you look at it, the way you package apps on Apple, that monolithic icon hides a huge number of stuff, including all the libraries. They keep it all in the library, in the icon. And so they're very large apps. I just downloaded an app that was half a gigabyte.

Steve: Ooh.

Leo: For something that didn't do anything. Because honestly, otherwise you'd have 18 copies of Electron in all different system folders. Instead, you have 18 copies of

> Electron, but they're all within the package icon. So maybe that's - I don't know if it's better, but it at least eliminates DLL Hell.

Steve: Well, so one of the things that Windows did, because the concept was an app didn't have to have its own DLLs, it would be able to use them if they were in the system. So that allowed the Windows loader to go looking around for them. And that's been a cause of problems in the past. Turns out, believe it or not, even now that hasn't gone away. So in the case of Trend Micro Maximum Security versions below v16.0.1221, there's a vulnerability that just got fixed. The researchers found that the service known as "coreServiceShell.exe," it loads a DLL, paCoreProductAdaptor.dll. However, in the case of a missing DLL, the lack of safe DLL loading, and the lack of DLL signature validation, hackers are able to exploit this as a security hole to cause their own unsigned malicious replacement DLLs to be loaded by the Trend Micro maximum security service every time that service runs.

And the service runs with the NT AUTHORITY\SYSTEM permission, which is the highest set of permissions in the system. Basically it's like kernel root-level permission. So it turns out that it is extremely easy to evade all the other provisions that Trend Micro put into place in order to prevent their system from being hacked because they weren't checking the signature of the DLL that they were loading. Windows would just go find a DLL of the proper name. And all the bad guys have to do is arrange to get that DLL in the Windows search path upstream of where Trend Micro put the real one, and that would cause a persistent, highly privileged load of a malicious DLL. So as I said, this whole DLL thing, not only do things not work, but it has been a serious security problem moving forward.

The second vulnerability in Kaspersky Secure Connection is a VPN client, part of Kaspersky's Internet Security solutions, which is used - the VPN is used to create a secure connection back to the vendor's servers for transferring updates and information. Like Trend Micro, their VPN versions below 4.0 does not check the signatures of the DLL it loads. And it also runs that DLL with maximum system permissions. And because it's a DLL, it's possible for the bad guys to drop a malicious one elsewhere in the system, in the Windows DLL search path, which Kaspersky will cause to have loaded and executed preferentially to their own, thanks to Windows.

And Autodesk, not very much different. In this case, all of these things are loading with the NT AUTHORITY\SYSTEM permission, and Autodesk was found not to be checking the signature of its DLLs. The researchers wrote: "After an attacker gains access to a computer, they might have privileges limiting them to access only certain files and data. But the Autodesk service provides them with the ability to operate as NT AUTHORITY\SYSTEM, which is the most powerful user in Windows, so they can then access almost every file and process which belongs to the user on the computer."

So what we've seen is that the DLL design pattern and methodology is so deeply ingrained into Windows that there really is nothing Microsoft can do about it today. The fact that that Windows side-by-side mess doesn't directly address DLL security, it only is there so that modern Windows has any chance of running at all these days. I was initially a bit surprised to learn that processes running with maximum permission were allowed to have unsigned DLLs loaded into their process space. But upon reflection, changing that at this point would also break too many longstanding assumptions. So I suppose the best we can do is to keep patching and be glad that there are folks like SafeBreach Labs, these guys, who take it upon themselves to find and responsibly disclose those problems when they are found.

And I wanted to take a moment just to sort of touch base on the Joy of Sync solutions that I found. It occurred to me because I stumbled upon some of the notes that I had

made, and I realized that in all the time that I have been using it, I have never once had a problem. It has never once failed me. That is, it is essentially the reason I went looking for something else when I was having problems with Dropbox's sync. So I don't know what the magic potion is, the magic programming dust that Sync.com has, but they have it.

And so I just, you know, as we know, it's often the case that we'll start to use something, and then something will happen, and we'll go, oh, well, and stop using it and fail to let everybody know that our recommendation changed. So I just wanted to remind people about it. They make 5GB free for anyone who is interested. I have a link in the show notes which will give you an extra gig free if you want to give it a try.

So you start off with 6GB free. It's just grc.sc/sync. And if you use that link, it expands to a link which actually both of us, you and I, get a gig free. I already have 3TB, so I'm fine. But I figured, hey, difference between five and six gigs, that's more significant. So anyway, I am every bit as happy and bullish with Sync.com as I was. I don't know if - I think I did the Joy of Sync podcast before I left for my tour.

Leo: Yeah, you did, yeah.

Steve: Because I think I told you, Leo, that I had forgotten to put the presentation on my laptop. And it was like, oh, crap. But then I realized, I mean, we were literally on the tarmac, and I said to Lorrie, I said, "You know, I think I forgot to put the presentation on the laptop." And she looked at me with alarm. And I said, "But it's okay because I do all of my work now underneath my Sync folder." And so I knew that I would be able to get it wherever I was.

So anyway, grc.sc, that's for shortcut, grc.sc/sync. And again, my recommendation stands. I think these guys have nailed it. And I'm also, for other applications where I want to do my own peer-to-peer, Syncthing. I have been using it also extensively for other purposes where it doesn't make sense or there's no need to sync to the cloud. And Syncthing has never let me down. And I have it running on my Drobos. It's keeping my two Drobos in sync, as well. So anyway, I just wanted to touch base.

Leo: You mentioned that Sync.com - I wanted to use Sync.com because it's end-to-end encrypted, which is awesome.

Steve: Yes.

Leo: But they don't...

Steve: Yes, in fact...

Leo: Go ahead.

Steve: Go ahead.

Leo: Well, they don't have a Linux client.

Steve: Yes, yes. That is the one problem. They don't have Linux. They understood after we talked about it on the podcast, and a lot of our listeners said, hey, thanks, we'd love to use it, but without Linux - and so anyway, they did receive that message.

Leo: I hope they make one. I, because of that, ended up using pCloud, which is not end-to-end encrypted, but they have an end-to-end encrypted folder. So you can say this folder I want always end-to-end encrypted. Which for my purposes was actually sufficient. I'm not too worried about it. They also had, and I like this, a lifetime price. I think it was like 200 bucks, and that's it, for 2TB. So I just bought the 2TB for life. And I just use it as - in some ways I'm wondering if people really need NASes much anymore. As storage gets cheaper in the cloud...

Steve: I agree.

Leo: For me it ends up being much more convenient than my Synology NAS. I just keep everything - I still run the Synology NAS so I have a local backup.

Steve: I agree. Yup.

Leo: But, you know, I think it's a pretty nice solution.

Steve: Well, and remember that at least the Sync.com also has full versioning. You can go back to any prior...

Leo: Right, as does pCloud, yeah.

Steve: Okay, yeah, go back to any...

Leo: But Dropbox does that now, too.

Steve: Yeah.

Leo: Now, I love that because, if you screw up, you can go back in time. I'm actually pretty happy with pCloud. And they do have a Linux client. And that's, for me, half of my computing's on Linux, so I kind of have to have that.

Steve: So I found the reference.

Leo: StrandHogg.

Steve: Yes. Promon, who we'll be talking about, calls this vulnerability StrandHogg, which is Old Norse, referring to the Viking tactic of raiding coastal areas to plunder and hold people for ransom.

Leo: Oh, that's the Strand, probably.

Steve: So I guess they weren't - I guess those Vikings weren't all just glamorous and...

Leo: Oh, it wasn't all mead and beer in Viking horns. No, no.

Steve: So these guys, Promon security researchers, have found proof of a dangerous Android vulnerability which, as we know, they call StrandHogg, that allows real-life malware to pose as legitimate apps, with users unaware they're being targeted. They are a partner with the security firm Lookout, whom we've talked about several times. Lookout confirmed that they've identified 36 malicious apps exploiting the vulnerability. And among them were variants of the BankBot banking trojan, which was first seen in 2017. During testing, Promon researchers found that all of the 500 most popular Android apps, as ranked by the app intelligence company 42 Matters, are vulnerable to StrandHogg. All versions of Android are affected, including Android 10.

Leo: Is this their logo, or did you make this up?

Steve: Yes, yes. No, that's their logo. In fact, there's one on their page that has that mask on top of one of the Android bots.

Leo: Oh, my.

Steve: There's two Android bots on either side, and then that one in the middle is the evil Android bot. Anyway, BankBot - that's the malware that is being used - is one of the most widespread banking trojans around, with a ton of variants, and new ones appearing continually. BankBot attacks have been detected around the world, in the U.S., Latin America, Europe, and the Asia Pacific region. StrandHogg was first detected when banks reported that their customers were reporting missing funds from their accounts.

So in their little top-of-the-page summary, they summarize what's the impact of their finding. All versions of Android are affected, including Android 10. All top 500 most popular Android apps can be abused. Real-life malware is exploiting the vulnerability. That is, this is in the wild, doing this now to people. Thirty-six malicious apps exploiting the vulnerability have been identified. Oh, wow.

Leo: Bless you.

Steve: Excuse me. Yeah, I don't think I've ever sneezed before. I've come close a couple times, but there it is. The vulnerability can be exploited without root access. So all it takes to do this is to get a piece of malicious software into an Android device. And unfortunately, Google is always removing malicious code from the Google Play Store. When this is exploited by hackers, they can listen to the user through the microphone.

They can take photos through the camera. They can send and receive SMS messages, make and/or record phone conversations, phish login credentials, get access to all private photos and files on the device, get location and GPS information, get access to the contacts list, access the user's phone logs, I mean, it's like they were somehow able to root your device, but it turns out that's not necessary.

So what's going on here? How does this happen? It turns out it's the harvesting of dangerous permissions. And I have a graphic from their report in the show notes. They show the user clicking an app icon of a legitimate app in order to launch it. But instead of seeing the legitimate app, the malware is displayed, impersonating the app, which then asks for any permission while pretending to be the legitimate app, like allow to send and view SMS messages. Which, depending upon the app, might be something, a permission you would expect to give the app.

But unfortunately, as a consequence of leveraging this - and here's the key - designed-in aspect of Android, you're actually giving the permission to the malware, not to the legitimate app. Which is then allowed to run and take advantage of the victim. So the vulnerability makes it possible for a malicious app to ask for permissions while pretending to be the legitimate app, and an attacker can ask for any permissions they want: SMS, photos, microphone, GPS, text messages, view photos, eavesdrop, track the victim's movements, whatever. The attack can be designed to request permissions which would be natural for different targeted apps to request to reduce the suspicion on the part of the victims; right? So that would be natural in a little bit of a social engineering hack. Users are unaware that they are giving permission to the hacker and not the authentic app they believe they're using.

But wait, there's more. Powerful credential stealing apps are also enabled by the presentation of faked login prompts. You click on an app where you are expecting to need to log in. A malicious login page is displayed on the victim's screen instead of the legitimate app. Then those credentials are supplied to the actual app in order to log in. Basically, it's a man-in-the-middle attack on your device.

Leo: Is this the screen overwrite permissions that we've talked about before?

Steve: Yes. It is an extension of those longstanding screen overwrite permissions.

Leo: It's always been a problem. But Google seems loathe to take them out, I think because apps like Facebook use it for their Messenger pop-up bubbles and things like that.

Steve: That's exactly right. So what's going on? What makes StrandHogg unique is that it enables these sophisticated attacks without the need for a device to be rooted. It leverages what are arguably weaknesses inherent in Android's multitasking system to enable powerful attacks that allow malicious apps to masquerade as any other app on the device. The exploit uses an Android feature called Task Affinity, which allows any app, including malicious ones, to freely assume any identity within the multitasking system they desire. And as I mentioned at the top, Promon's research found that all of the top 500 most popular Android apps are vulnerable to this attack across every version of Android.

So what's interesting here is that their research significantly expands upon research which was carried out by Penn State University and the guys at FireEye in 2015. Back then, the researchers theoretically described certain aspects of the vulnerability. At the

time, Google dismissed the vulnerability's severity. But Promon now has clear evidence that hackers are exploiting StrandHogg to gain access to devices and apps. The specific malware sample that Promon analyzed did not reside on Google Play, but it was installed through several dropper apps and hostile downloaders distributed on Google Play.

These apps have since been removed. But in spite of Google Play's Protect Security Suite, dropper apps continue to be published and frequently slip under the radar, with some being downloaded millions of times before being spotted and deleted. And just due to the nature of the way Google Play works, Google is necessarily reactive. Of course they will remove anything that they become aware of. But this means that some people are downloading and being hurt by malicious apps in the interim. For example, there was one CamScanner which was a PDF creator that contained a malicious module. It had been downloaded 100 million times before Google realized that it was malicious and pulled it.

So Promon's CTO, their Chief Technology Officer, Tom Lysemose Hansen, he commented: "We have tangible proof that attackers are exploiting StrandHogg in order to steal confidential information. The potential impact of this could be unprecedented in terms of scale and the amount of damage caused because most apps are vulnerable by default and all Android versions are affected."

So I wanted to dig a little bit deeper into this. I was curious about what are the details. So I went back, and I found the original security research which was presented at the 2015 USENIX Security Symposium back in August of 2015. The researchers titled their paper "Towards Discovering and Understanding Task Hijacking in Android."

And in their abstract they wrote: "Android multitasking provides rich features to enhance user experience and offers great flexibility for app developers to promote app personalization. However, the security implication of Android multitasking remains under-investigated. With a systemic study of the complex tasks dynamics, we find design flaws in Android multitasking which make all recent versions of Android vulnerable to task hijacking attacks. We demonstrate proof-of-concept examples utilizing the task hijacking attack surface to implement UI spoofing, denial of service, and user monitoring attacks. Attackers may steal login credentials, implement ransomware. and spy on users' activities.

"We've collected and analyzed over 6.8 million apps from various Android markets. Our analysis shows that the task hijacking risk is prevalent. Since many apps depend on current multitasking design, defeating task hijacking is not easy. We've notified the Android team about these issues, and we discuss possible mitigation techniques in the paper."

So this is not news to anybody at Google working on Android. And then I dug a little deeper. I was worrying about this "allowTaskReparenting" attribute. So it's in the Android 1.6 SDK. And it's called allowTaskReparenting. And as an example, they said: "If an activity" - and this is from AndroidCookbook.info. "If an activity has its allowTaskReparenting attribute set to 'true,' it can move from the task it starts in to the task it has an affinity for when that task comes to the fore." And as you can hear, that sounds like exactly what this malware has been designed to do. It creates an affinity bound to some task it wants to impersonate, turns on allowTaskReparenting, and then automatically gets switched into that task's process.

They said: "For example" - and this was the example they give - "suppose that an activity that reports weather conditions in selected cities is defined as part of a travel application. It has the same affinity as other activities in the same application, the default affinity, and it allows reparenting. One of your activities starts the weather reporter, so it initially belongs to the same task as your activity. However, when the

travel application next comes forward, the weather reporter will be reassigned to and displayed with that task."

So in other words, this has been designed to do this. There is something designed into Google which, if malware is allowed to be installed in someone's device without itself needing any permissions, without itself needing any deep access, is able to look around the phone, see what's there, create these affinities, and then impersonate by doing a UI layer, exactly as you said, Leo, a UI layer over the real app in order to obtain the permissions for itself. And that is now being actively exploited in the Google Play Store for any new malware which is installed and does this, until it is identified as doing so and then removed. And then of course another instance of it is installed under some other name.

So I don't know what Google's going to do about this. They know about it. Maybe the fact that this report is now public as of yesterday, and this is now getting some real scrutiny, maybe they have some means in the works to fix it, but haven't implemented it yet. But this if nothing else will turn up the heat on them, and that's just all to the good.

Leo: We've seen this before, and they didn't face it last time, as you mentioned, four years ago. I think it's because there are major apps like Facebook Messenger, which is you don't use Android, but there's a chat bubble feature which I really like. If you get a Messenger chat, a little head will show up in a bubble on top of the screen that you can move around, you can tap, you can interact with. So it is part of its multitasking capability. And I suspect they don't want to turn it off because they don't want to get Facebook angry.

Steve: Yeah.

Leo: I notice that's not available on Apple. You can't overwrite the screen.

Steve: Nope.

Leo: Nope. There's the peril involved, though. I'm always, you know, I love Android, and I'm very happy with my orange Pixel 4. I just don't download apps at random; you know? I think that's the best thing to do.

Steve: Correct. That is all the user can - yes. That is our takeaway advice, not for the first time on this podcast, is just, if you can, stick with well-known mainstream - and don't just download everything that's available because it's free.

Leo: Right. You might be tempted by that PDF scanning app. But get the one from Google. Don't get the one from [crosstalk] or whatever it is.

Steve: Right. And the fact that 100 million people have downloaded it doesn't mean that it's safe.

Leo: That's amazing. Wow, that's amazing; isn't it?

Steve: Yeah.

Leo: Terrifying. Now, you'd have to be targeted in this attack. Or no?

Steve: Well, no. No, no, no. And if this thing doesn't know who you are, it will find out who you are.

Leo: And then contact the servers and say, hey, I have access to Leo's phone. Would you like to see what he's doing right now?

Steve: Yep, exactly.

Leo: Okay. Geez, Louise. Steve Gibson does it again. And all I can say is, if you don't listen to this show, your hard drive could die. That's all. I'm just saying. You've got to listen every week to Security Now!. You don't have to listen live, although we do do it live every Tuesday, 1:30 Pacific, 4:30 Eastern, 21:30 UTC. Figure out what your time is. And go to TWiT.tv/live. There's audio and video so you can listen while you work. Boss will never know. It's good, actually the boss should encourage it. It's a good way to keep up on all the perils, risks, and inside and outside baseball stuff that has to do with technology.

You can also, if you are listening live, chat with us at irc.twit.tv, always a very active chatroom during the show. You can get on demand versions of the show. Steve's got kind of two unique versions. He has a 64Kb audio file, but he also has a 16Kb audio file. So if you're on a bandwidth-limited connection, that's a great choice. It's a little scratchier, but at least you get it. He's also got the smallest version of all, which is the fantastic transcripts that Elaine Farris does for him. So you can download and read along as you listen. Those are all available at GRC.com.

While you're there, Steve didn't mention it this week, but pick up a copy of SpinRite. That's his bread-and-butter and the world's finest hard drive maintenance and recovery utility. SQRL is official. You can find out more about it, learn about the API, how you would write a SQRL - I want somebody to work on a SQRL authentication plugin for our TWiT Forums at Twit.community. It's a Discourse server. And I think that Discourse is widely used. You already have it for XenForo, and people that use your forums love it.

Steve: And also for WordPress. There is a WordPress plugin.

Leo: Yeah, yeah.

Steve: So we need Disqus.

Leo: Yeah. So Disqus would be great. I use Disqus. I would use it on my WordPress blog. But I use Disqus as the commenting engine on it.

Steve: Right, right.

Leo: And maybe I should just go to native WordPress commenting, and then I could do it. I just don't want to lose all the existing comments. I'll look tonight and see if I can export the Disqus stuff into WordPress because I don't want to use Disqus anyway. It's another person watching what you're doing and all of that.

Steve: Yeah.

Leo: Yeah. Besides SQRL and SpinRite, there's all sorts of other stuff like ShieldsUP!, lots of free software. It's a rathole that you'll spend many happy hours perusing: GRC.com. He's on Twitter at @SGgrc. And that would be a good place if you have a question or a comment or want to leave a message for Steve. If you've got some inside dope, you could direct message him at @SGgrc on Twitter. He does not yet have his own Minecraft server. That's just a matter of time, though. No. It's never going to happen. We have on-demand audio and video of every show, as well. Video, yeah. So you can see us as well as listen. And that's available at TWiT.tv/sn. TWiT.tv/sn.

Best thing to do, though, honestly, subscribe. Then you don't even have to think about it. You'll just get it automatically, audio or video, the minute it's done. And if you are subscribing, it helps us because that way we kind of rise through the ranks in the various podcast applications. We're more likely to show up in the discovery tab. So whichever podcast application you use, please do subscribe to Security Now!.

Steve, you have said this off the air, but I'm going to say it on the air. In a couple of weeks, the Christmas Eve show.

Steve: Ah, yes.

Leo: I can't wait. This is going to be a special episode. You want to tell us about it?

Steve: Well, I'm noticing that we're starting into a new decade with 2020. So I thought it would be fun to do - I know.

Leo: Hard to believe.

Steve: I thought it would be fun to do a decade of hacks, looking back over the last 10 years of hacks and attacks, and just sort of do a little retrospective of everything that we've covered. So we'll deal quickly with any news of the week, and then spend most of our time looking back over the past 10 years.

Leo: What a decade it has been for security. That'll be the December 24th edition. We are recording it the day before. We didn't want do it on Christmas Eve, so we're recording that at our usual time, but on Monday the 23rd, just in case you watch live. And then the following week the New Year's Eve Security Now! will be a best-of episode from all the episodes all year long. And we love those best-of episodes. They're always a lot of fun. So Steve, we'll see you right back here next Tuesday.

Steve: Okay, my friend. Till then, thanks. Bye.