TRRespass

Description: This week we look at a new unpatched zero-day attack affecting billions of Windows users, Mozilla's reversal on TLS 1.0 and 1.1 deprecation due to the coronavirus, a welcome micropatch for Win7 and Server 2008, Chrome's altered release schedule during the coronavirus, Avast's latest screw-up, a new threat affecting Android users, the results from last week's Pwn2Own competition, and a few observations about the coronavirus math and some worthwhile explainer videos. Then we look at where we are with Rowhammer after six years.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-759.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-759-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson's feeling better. We've got a great show for you. He is going to talk a little bit about COVID-19, but then we're also going to talk about the zero-day exploit that affects all versions of Windows and how you can fix it, and why Rowhammer is not about to go away anytime soon. It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 759, recorded March 24th, 2020: TRRespass.

It's time for Security Now!, the show where we cover your security, your privacy, talk about computing. It's really a great show for anybody who's interested in these matters. And of course it's all because of this guy, Steve Gibson, the man about town. He's in a top hat and tails today with a little - because he's feeling better; right? You're feeling better.

Steve Gibson: I'm a little less, well, yes. The fact that I'm here. I may, like that, cough a little bit.

Leo: Do it in your sleeve. That's okay.

Steve: Yeah. The thing that's enduring is a very, like the tail end of just a dry, non-productive cough.

Leo: Right.

Steve: So there is a respiratory component which never got very bad for me, fortunately. But that seems to be hanging on. But I've got most of my energy back.

Leo: Good.

Steve: Although I have to say, Leo, I got knocked on my butt after last week's podcast.

Leo: Oh, really. Sorry.

Steve: Oh. Lorrie looked at me, and she said, "Okay, honey, now remember this when you're considering doing it again." And I said no.

Leo: That took it out of you. That took it out of you.

Steve: It really did.

Leo: Oh, I'm sorry.

Steve: But I'm glad I did it, and I'm feeling really back up to snuff. Well, it was my decision. I could have said I, you know, can't do it. But I don't want to break my record, Leo.

Leo: Well, it's unblemished, despite illness.

Steve: We have, for our 759th podcast, "TRRespass" is the title, with two R's, T-R-R-e-s-p-a-s-s. We're going to essentially take a look at where we are six years downstream from the Rowhammer revelation.

Leo: Wow. It's that long? It's been six years?

Steve: Half this podcast ago.

Leo: Wow.

Steve: Yes. So we're going to take a look at a new unpatched zero-day attack, found in the wild, being currently used in limited fashion against billions, well, there are billions of potential Windows users, but it's currently being selective. Microsoft knows about it. They have not patched it. We have got to wait three weeks, which is when the April Patch Tuesday occurs.

Leo: Are you talking about the type meta? The type flaw? The zero-day with type rendering?

Steve: Yes, yes, yes. The Adobe font rendering attack. There are some mitigations we'll talk about. You know, and so, well, anyway, we'll get to that. Also Mozilla reversed themselves on TLS 1.0 and 1.1 deprecation.

Leo: Good, I think.

Steve: Yeah, we're going to take a welcome look at a micropatch for Win7 and Server 2008 for things that are maybe important, which Microsoft is never going to fix. So we're beginning to walk down this road now of encountering important things which are continually being discovered in a version of Windows that still has not now half, but still close to half, of the user base. We've also got Chrome's and Edge's altered release schedules. Avast's latest screw-up. We have a new threat affecting Android users. The results of last week's virtual Pwn2Own competition.

Leo: That was pretty funny, yeah.

Steve: Yeah.

Leo: They mailed in the exploits.

Steve: We have a few observations about the coronavirus math that I wanted to point our listeners to, or just some things to think about. And also I'm going to share two more of my shortcuts for two really amazing videos that I've discovered since last week. You've seen them both. I think I've sent them both to you, Leo.

Leo: Yes, yes.

Steve: And then we're going to dig into Rowhammer, where we are after six years.

Leo: Wow.

Steve: And of course we have an apropos, very nerdy Picture of the Week.

Leo: We do indeed.

Steve: So a listener of ours by the name of Tony Davis emailed this Picture of the Week to me, saying he thought it would be perfect, given the current state of the world. And I have to agree. It is a sort of a sign that reads: "Please stay @ 127.0.0.1. Don't be 255.255.255.255."

Leo: Now, I know 127.0.0.1 is home, local home.

Steve: Correct. Please stay at home.

Leo: What is 255, quad 255?

Steve: So all Ethernet controllers and Internet-connected devices will accept - they will listen to anything coming from there. That's the broadcast IP.

Leo: Oh, I get it.

Steve: So it's perfect. It's like, don't be broadcasting whatever it is you might have.

Leo: Please stay at home. Don't be broadcasting. I love it.

Steve: Exactly. It's perfect.

Leo: Very clever.

Steve: So thanks for thinking of us, Tony. And I did not have this in the show notes. I stumbled upon my note about it after this had all been put to bed. But I wanted to let our listeners know, for those who have certainly kids and maybe young adults at home, Amazon has created something known as "Audible Stories." They said: "For as long as schools are closed, we're open. Starting today, kids everywhere can instantly stream an incredible collection of stories, including titles across six different languages, that will help them continue dreaming, learning, and just being kids. All stories are free to stream on your desktop, laptop, phone, or tablet." They said: "Explore the collection, select a title, start listening."

So basically they're helping the parents of stay-at-home kids by saying we're going to make our audio book collection available for kids at home. So give them something to do, which I think sounds like a great thing. So just wanted to mention that.

Okay. So you correctly guessed what the first issue of the day was, which are two new unpatched zero-days affecting billions of Windows users. Their advisory, Microsoft's advisory was published just yesterday on the 23rd of March, titled "Type 1 Font Parsing Remote Code Execution Vulnerability." They said: "Microsoft is aware of limited targeted attacks that could leverage unpatched vulnerabilities in the Adobe Type Manager Library, and is providing the following guidance to help reduce customer risk until the security update is released."

They said: "Two remote code execution vulnerabilities exist in Microsoft Windows when the Windows Adobe Type Manager Library improperly handles a specially crafted multi master font, Adobe Type 1 PostScript format." They said: "There are multiple ways an attacker could exploit the vulnerability, such as convincing a user to open a specially crafted document or viewing it in the Windows Preview pane."

They said: "Microsoft is aware of this vulnerability and is working on a fix." Also they're aware of attacks in the wild, thus it's a zero-day. They said: "Updates that address security vulnerabilities in Microsoft software are typically released on Update Tuesday,

the second Tuesday of each month. This predictable schedule allows for partner quality assurance and IT planning, which helps maintain the Windows..." blah blah blah blah.

So basically they're saying we're sticking to Patch Tuesday for this. Coincidentally, and unfortunately, this is one of those months which has the latest possible Patch Tuesday, where the 31st of March is next Tuesday, so the first is the following Wednesday, and the second Tuesday is as far back as it could be. So three weeks from today, essentially.

Both of the unpatched flaws are known to be used in limited targeted attacks, and they impact all supported versions of the Windows operating system, I guess because the Adobe Font Type Manager Library has been there through all of them. Windows 10, 8.1, Server 2008, 12, 16, and 19, and Windows 7, which of course we know for which Microsoft ended their support, as they did for Server 2008, in January.

The vulnerabilities reside in this Adobe Font Type Manager Library, which is a font-parsing and display subsystem used by Windows Explorer to display the content of a file in the Preview or the Details panes, without users needing to open it. So this is potentially a "you don't have to do anything" exploit. It's also used by many pieces of third-party software. It's part of Windows. So third-party software can just presume that this library will answer the call. The problem is that the Type Manager Library is improperly handling a specially crafted multi master font in the Type 1 Postscript format, and it allows remote attackers to execute arbitrary malicious code on target systems.

Leo: Who was it said "Interpreters are hard"? Oh, yeah, Steve Gibson.

Steve: Ah. Yes. So the obvious attack route involves convincing a user to open a specially crafted document or viewing it in the Windows Preview pane. It's not clear whether the flaws can also be triggered remotely over a web browser by convincing a user to visit a web page containing specially crafted malicious OTF fonts. And there are multiple ways an attacker might be able to exploit the vulnerability. And interestingly enough, Microsoft mentions the WebDAV, the Web Distributed Authoring and Versioning client service, as another vector into the system. So it's clear we're not going to get anything...

Leo: Microsoft's implementation of WebDAV, or WebDAV in general? Because WebDAV is widely used.

Steve: Right. And so it's just that WebDAV is a way in to get to this faulty library.

Leo: Right. Well, there's lots of ways to get to the library. That's easy.

Steve: Yeah. So they are offering some workarounds. I like the final one, which is final in several senses.

Leo: Install Linux; right?

Steve: That's the Leo solution.

Leo: I swear to god, I don't run Windows on any machines anymore. It's just ridiculous.

Steve: No, no.

Leo: It's ridiculous. This is all versions of Windows, too; right?

Steve: Yup, all versions.

Leo: Even Windows 7. All versions. Windows 10.

Steve: Yes. And Microsoft knows of it. It's being used in targeted attacks.

Leo: Zero-day, it's a zero-day.

Steve: It's a zero-day. And we also know that zero-days are kept quiet and sneaky until they are discovered, at which point they then change their tactic. They want to get everybody they can before the patch shuts them down. So yes, Microsoft, when you posted this news, it was used in selective, targeted, stealthy attacks. Now it's probably being sprayed because they want to do as much, they want to get as much action out of this before it is shut down.

Leo: It's now or never, yeah.

Steve: And again, Microsoft is using apparently the fact that it is in targeted attacks to say, well, you know, three weeks, that's not going to be so bad. Good luck. So it is possible to set - there is an option under the View settings of Windows Explorer, you know, the thing that we use for viewing our files in Windows. Not Internet Explorer, Windows Explorer, or just Explorer. You can turn on a checkbox, "Always show icons, never thumbnails." So if you enable that, what that does is that prevents Explorer from going out to Adobe and asking for its help in rendering a thumbnail to show you in the Preview pane or in the Summary pane. So you can turn that on if you want to thwart this one avenue in.

However, they also recommend disabling the WebClient service under Windows, and you could do that by going to the Windows Services, scrolling all the way down to W's, where you'll find Windows WebClient. And mine was set to "manual" and "auto trigger." They want you to set it to "disabled" so nothing will start it. And so that helps you solve like a different entry into the system.

Leo: Nobody's going to - nobody except you and people who know what they're doing is going to know about this or do it.

Steve: Yeah, yeah. Well, and that's the problem is that, you know, our listeners, who listen because they like these little kinds of...

Leo: Yeah, this is why you listen, yeah.

Steve: Exactly. They'll be safe. But the rest of the world is now subject to three weeks of probably an escalating attack using this.

Leo: You couldn't pick a worse three weeks, of course, because everybody's working from home. They took those Windows machines home. If I were a bad guy, you know what I'd do, I'd take advantage of that time, get on as many systems as possible, so that when they're brought back to work, you could really get -you could have some fun.

Steve: Well, also we are seeing a sadly predictable huge upswing in coronavirus-leveraged attacks. You know, so...

Leo: Oh, yeah. People are awful. People are awful.

Steve: I know. It's just unbelievable. We talked last week about some actual DDoS attacks on medical infrastructure. I did see one little note saying that some of the ransomware people were going to lay off the health services industry during this time.

Leo: Oh. Yeah. I hope that's universal. No. Geez.

Steve: Anyway, so the problem would be that, if this can be leveraged in a social engineering attack, this is a time when lots of people are very worried about this problem and are thus more subject to not thinking before they click. And so as we've often said, we haven't used this little pithy bit of advice for quite a while, it's never download anything that is offered to you.

Leo: Period, yeah.

Steve: That's a golden oldie. Period. If something says, oh, you need to update your Flash Player, no. Never download something that is offered. Only go and get it yourself, if you have reason to believe you need it, because first of all, everything seems to be working just fine without this thing that you're now told you need. So just blow it off. Just no.

Okay. But just to finish, in the show notes I've got Microsoft's good advice, which they save to last because it's the most onerous. And that is to rename this DLL. It's atmfd.dll. And there are two scripts, one for 32-bit systems and a double-length one for 64-bit systems. It switches you - you use it from the command prompt. I'm sure you have to be an elevated command prompt, an admin command prompt. Switches you to the Windows directory system32. Then runs the "takeown" command to take ownership of that DLL. Then runs the access control list utility, icacls.exe, saving the existing access control list for atmfd.dll into a temporary file. Then grants admin's access to the DLL. Then renames, which you are now an admin, renames atmfd.dll to x-atmfd.dll.

Basically, that just means that no one in the system that expects to be able to simply load this Adobe Type Manager DLL will then succeed. So that forecloses all access to it.

Again, limited targeted attacks. It's unlikely that any individual that we're talking to will get hit by this. But in the interest of caution, presumably in three weeks the update will replace this DLL with the right one. So the old one, the x-atmfd.dll - oh, and on 64-bit systems you have to do it twice because the 64-bit systems have the old 32-bit and the new 64-bit versions. So it does it twice. But it's quick to do. And if I were concerned, that's what I would do. It just removes this Adobe Type Manager DLL from your system by renaming it to something that will never be seen. And then in three weeks the update will replace that empty slot with a fixed atmfd.dll.

Leo: Now, do we blame Adobe or Microsoft for this?

Steve: I don't know. And maybe that explains the delay. Maybe if this were Microsoft's own code, they'd be able to say, ooh, crap, and jump on it immediately and fix it. It may be that, because this thing is actually an Adobe problem, they've had to work through just the intercorporate stuff; and Adobe said, well, we'll work on it and get it to you in a couple weeks.

Leo: So these Type 1 fonts are an Adobe font style and usually on Windows.

Steve: They're the early, early fonts.

Leo: They're early, right. Then TrueType took over. And so you see those OTF fonts are the older ones, and TTF are the newer ones. And the Type Manager was necessary, I guess, for rendering the fonts. But it might be a Microsoft product that's labeled Adobe Type Manager because it's for managing the Adobe Type 1 fonts. So it's unclear; right? I mean...

Steve: Yeah. And I was about to go click on it and check the properties because...

Leo: Maybe not.

Steve: We could see where it came from. Except that, even if it came from Adobe, Microsoft might take ownership of it and wrap it in their own shell.

Leo: So you're looking for atmfd.dll; right? If you delete that, you're good; right?

Steve: Yeah.

Leo: Okay. And there's two of them? Is there one in 32 that says atmfd32.dll?

Steve: Yeah, well, actually they're in different directories. So they're all atmfd.dll.

Leo: I just [crosstalk] how Windows is made. Oh, my god.

Steve: It's, well, they've had this really awkward problem.

Leo: Just leave the 32-bit over there, yeah.

Steve: Yeah, and now we've got the Program Files x86, which was like - and then they decided, oh, we made a mistake with that, so we're going to abandon that approach.

Leo: It's just so ugly.

Steve: It's really become - yeah. It's really become a problem through the years, yeah. There is a registry tweak also in the show notes that applies to Windows 8.1 and earlier. So if you are a Windows 7 user, you could apply, just apply this registry tweak that should take it offline and out of service, and then that should be good also. It's a mess.

Leo: Yeah. It's in the show notes, but it's not as a download, it's an actual edit you have to do. Is that - oh, my god. See, no one's going to do this.

Steve: Yeah. Look at it. It's just a mess.

Leo: [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows] "DisableATMFD"=dword: - can you just write "1," or do you have to write "00000001"?

Steve: I think you can write "1."

Leo: It's just "true."

Steve: Yeah.

Leo: Geez, Louise. Okay.

Steve: Okay. So our listeners all know that exists for three weeks, then they're going to fix it. So you could rename the DLL, if you like. You could turn off some display things in Windows. Or not worry about it, who knows.

Leo: Who knows.

Steve: Microsoft - not Microsoft. I see a big "M," and I think Micro. Mozilla reversed itself on TLS v1.0 and 1.1 deprecation. Get this, Leo, due the coronavirus. As we know, we talked about it a couple weeks ago, TLS 1.0 and 1.1 was going to be dropped on March 10th, two weeks ago, with the release of Firefox 74 - which is the current release, it's what I'm using now - to improve the security, as we know, of website connections so that sites not supporting the better protocols 1.2 or 1.3 would then start showing a blank

page. I mean, the big scary warning "Secure Connection Failed" error page. I mean, it was going to be - we're not just saying "not secure" anymore, we're saying no. And remember the plan was nobody's going to make this change until they're forced to. The U.K. firm, can't remember their name, Netcraft, their survey showed that this month there were still 850,000 websites not yet offering 1.2 and 1.3, still only offering 1.0 and 1.1.

But then coronavirus happened. And I've got a link to their release notes, and their release notes show, they say: "We have disabled TLS 1.0 and TLS 1.1 to improve your website connections. Sites that don't support TLS version 1.2" - and they meant 1.3 - "will now show an error page." That is crossed out in the release notes for Edition 74 of Firefox.

Leo: Oh, and I love the reason why.

Steve: Yes. "We reverted the change for an undetermined amount of time to better enable access to critical government sites sharing COVID-19 information."

Leo: Oh, my god.

Steve: Uh-huh. So essentially, coronavirus just tipped the balance enough that apparently some government sites were among those 850,000 that have not yet gotten their act together and moved to 1.2 or 1.3. And Mozilla was realizing that their users, their Firefox users, were getting connection failure when trying to go to those sites. So they said "whoopsie" and backed out and are leaving 1.0 and 1.1 on.

To check that, and any of our listeners can, SSL Labs not only has the famous server-side check, but they've got a browser client-side check. So you can go to SSLLabs.com and select, I think it's the second one down on the right, the client test, which will quickly check your client for which protocols it supports. And I went there with my Firefox 74, and it showed it supported 1.3 and 1.2, and it was all happy about those. And, oh, it supported 1.1 and 1.0, which was red, it did not like the fact that my browser still supported those, but it showed yes, they are supported. So at some point, presumably, you know, who knows when.

And I was wondering whether, and I didn't have a chance to check this, whether Chrome also reversed themselves because this was all being done - remember Chrome and Mozilla and, well, basically Chromium, Mozilla, and Safari were all in lockstep with this. I hope everybody backed off, if there actually were sites, as presumably Mozilla discovered, that were not coming up if 1.0 and 1.1 were blocked. Anyway, so it's been reversed. And the pressure campaign will resume once it's no longer a health-critical issue for Mozilla's Firefox users.

Leo: That's interesting. Boy. I wonder what government sites are still using TLS 1.0 and 1.3? It must be 1.3; right?

Steve: Well, 1.1, hopefully 1.1. So, you know, it's not that those are obviously bad. I mean, it's not like they're broken, and you can actually decrypt traffic.

Leo: They're just not secure enough.

Steve: Yeah. They're not what we would want to be using moving forward. And presumably the sites are informational. They're not requiring high security credentials and so forth. So it's like, yeah, let people get there rather than not. Probably makes more sense.

Leo: Yeah. Worse to break the site in this case.

Steve: Yeah. Okay. So I think moving forward, and it'll be interesting to watch the slowly dwindling inventory of Windows 7 and Server 2008 machines on the Internet. But this notion of micropatching, you and I have talked about it a couple times. You've been a little hesitant.

Leo: Yeah, it makes me nervous, yeah.

Steve: Worrying about these guys.

Leo: Yeah. Who are these guys?

Steve: But if you look at them, they really do look like the real deal. So, okay. So where are we? On Patch Tuesday of this month we learned of the CVE ending in 0881. It's a remote code execution vulnerability affecting all versions of Windows. But of course Windows 7 and Server 2008 didn't get it.

Microsoft said: "A remote code execution vulnerability exists in the way that the Windows Graphics Device Interface (GDI) handles objects in memory. An attacker who successfully exploited this vulnerability could take control of the affected system. An attacker could then" - so, I mean, this is critical. This is a critical vulnerability in GDI. "An attacker could then install programs; view, change, or delete data; or create new accounts with full user rights. Users whose accounts are configured to have fewer user rights on the system could be less impacted than users who operate with admin rights."

Unfortunately we know that privilege escalation exploits are widely available also. So this should be, you know, should provide not much sense of refuse, or what I'm trying to say...

Leo: Refuge.

Steve: Refuge, thank you.

Leo: Or refuse.

Steve: "There are multiple ways an attacker could exploit the vulnerability. In a web-based attack scenario, an attacker could host a specially crafted website that is designed to exploit the vulnerability, then convince users to view the website." So in other words, just bringing up a web page. "An attacker would have no way to force the users to view the attacker-controlled content." Except, right, opening the web page. So an attacker

could convince users to take some action, typically by getting them to open an email attachment or click a link in an email or an instant message. "In a file-sharing attack scenario, an attacker could provide a specially crafted document file that's designed to exploit the vulnerability, and then convince users to open the document."

Okay. So right now this remote code, this critical remote code execution vulnerability has been present for two weeks, since March 10th, and it's never going to be fixed unless we micropatch. So we have, absent a micropatch, a potentially exploitable flaw in Windows 7, Server 2008, which Microsoft will never patch. If nothing is done, it will persist forever. It will get added to the cyberattack inventory of state actors and freelance hackers for use whenever they wish to leverage old and known, but never patched, Windows 7 and Server 2008 vulnerabilities. The point is we're now entering this era where there is going to be a growing inventory of ways to attack Windows 7 and Server 2008 that are never going to get fixed unless we go outside of Microsoft.

So the 0patch guys, again, it's numeral 0-P-A-T-C-H dot com, 0patch, and they call that "Micropatch," have generated another of their micropatches to fix the problem for those who cannot obtain the official fix from Microsoft. And of course anyone on the ESU, the Extended Services - ESU or ESR? I thought it was ESU. I don't know what that stands for. Maybe it's Service Releases, ESR. Anyone on that plan will still be getting them, which is what's galling for all other Windows users who don't have them. So it's available to 0patch's paying customers. The subscription is $26 per year per workstation. So, what, a little over $2 per month per workstation.

And in this case it repairs the memory corruption issue in Windows GDI+ by adding a block of code similar to the one Microsoft used in their official security fix. In other words, the 0patch guys are still getting the patches. They look at what Microsoft did, and they go, oh, yeah, okay. And they do the same thing and make it available to their customers.

Leo: Microsoft must not like that very much. Although...

Steve: Yeah. Suck on that, Microsoft.

Leo: Yeah, exactly, yeah.

Steve: On systems where the micropatch is present, it implements a logically identical check. Now, get this. If you were a subscriber a year ago, for example, at a little over two bucks per month per workstation, you're already fixed. That is, they have, 0patch has a little client module which is automatically updating your system as these micropatches become available, without you having to do anything.

Leo: What could possible go wrong with that?

Steve: You don't even need to reboot your system.

Leo: Oh, my god.

Steve: So it's way better than Microsoft Windows in that sense. So get this. On systems where it's present, it implements a logically identical check to Microsoft's, but also

records an exploitation attempt event before redirecting execution flow through the safe path. The patch for 32-bit systems is four instructions. The patch for 64-bit systems is five instructions.

And Leo, there's a YouTube video here you just might want to place into the stream. It's only a few seconds long, but it shows them doing the exploit, dropping an image on PowerPoint, and PowerPoint crashing when the micropatch is turned off. And then they flip the micropatch protection on, and they do the same thing, and it pops up a box saying that an exploit attempt was just caught. So it also gives you proactive information, if someone tries to hack your system.

So anyway, the next link is 0patch.com/pricing.html. And it's not very expensive. They have a free version. They give some of the things away for free. They're hoping to upsell people, obviously, to a subscription. And they're not asking for a lot. So again, Windows 7 people can decide if you want to sort of follow along with the podcast. For example, renaming the Adobe Type Manager DLL, just getting rid of it, essentially, for all time solves that problem under Windows 7. Doesn't look like there's a similar fix other than something like this micropatch. So that's what you would need to do if you wanted ongoing protection. So anyway, worth considering.

Leo: Are you doing this? Would you do this?

Steve: I'm not doing it; but I'm, well, yeah, I am still on Windows 7. I do have - we do know that they're continuing the operation of Windows Defender, so that's continuing to protect us on an ongoing basis. I'm glad for that. Lawrence Abrams is 100% bullish about these guys. He knows them. He's talking to them all the time. He's covering them in detail. And I think they're...

Leo: That's a good recommendation.

Steve: I think they're a legitimate, yeah, I think they're a legitimate outfit. In fact, I've been considering asking the 0patch guy to come on the show and talk to us.

Leo: You should, yeah.

Steve: And give us a sense for what they're doing.

Leo: Look into his eyes and see if he's trustworthy.

Steve: Exactly. And fortunately we can do that now virtually. Okay. So Chrome also. And since I put this together I discovered that Edge is following, because of course Edge is now Chromium based. Last Wednesday the 18th, Google announced that major Chrome browser and OS releases will be placed on hold due to the adjusted work schedules of employees having to work from home during the coronavirus sequestration. Their announcement stated that they would be placing priority on security, that is, browser security, which makes sense. As we know, most new browser versions bring us a mixture of new features, or often these days the removal of some creaky old features like TLS 1.0 and 1.1. And of course they also bring a bunch of security-related bug fixes.

So during this work-at-home time, the Google Chrome development team will be working remotely and prioritizing security updates that will be released as Chrome version 80 updates. So 80 is where we are now. And so Google will be doing incremental updates to 80, rather than jumping to 81 as had been planned. Google's Chrome 80.0.3987.149 was released right after the company announced that Chrome 81 was delayed, with security fixes which patched 13 high security vulnerabilities.

So we just got an update to 80, fixing 13 high security vulnerabilities, and don't hold your breath for 81. 81 is on the backburner. It was originally slated to start rolling out last Tuesday, on the 17th, but in the Google Developers blog they said we've decided that we're going to hold off on doing that. There was a bunch of stuff slated for release in that: modernized appearance, hit testing for augmented reality, app icon badge support, and initial support for Web NFC. We'll be waiting for that. We're just going to kind of hold on and just keep the existing Chrome secure.

Oh, and also affected were Android developers, sort of indirectly. They were informed, well, Android because of Google, they were informed that they could now expect to experience significantly longer than normal app review times due to the adjusted work schedules, as up to seven days or longer. A Google spokesman said: "Due to adjusted work schedules at this time, we are currently experiencing longer than usual review times. While the situation is currently evolving, app review times may fluctuate, and may take seven days or longer."

So anyway, the coronavirus, as we know, it's impacted Northern California, the so-called Silicon Valley, significantly, even though Google is spread out all over the place. Northern California was one of the first to implement the stay-at-home orders from California's governor. And actually I just heard this morning, I haven't seen any confirmation, Lorrie just told me that they're seeing some downturn finally. I mean, like the first indication of a downturn in the numbers in Northern California. So that would be amazing, if that turned out to be the case.

As we know, Avast has been having problems recently. They were behind one of the collisions with the Windows Update that we talked about a few months ago. And everyone knows, Leo, that you and I are both rather down on the idea of third-party AV. It's increasingly, dare I say, the cure is worse than the problem it's solving.

Leo: Be careful.

Steve: Yeah. We haven't heard much recently from Google's Tavis Ormandy. But earlier this month he used a nifty tool he developed three years ago, back in 2017. We've talked about it before. It allows Tavis to essentially excise Windows DLLs and run them under Linux, where automated fuzzing and other security tests can be performed sort of in vitro, as opposed to in vivo. It's a very cool concept. In this instance, this allowed Tavis to discover a trivially executed critical flaw in Avast's AV system. Tavis informed Avast; whereupon, after a week of scrambling around, they finally disabled this critical component of their AV product altogether.

What he found, what Tavis found, was a security flaw, a bad one, in Avast's JavaScript emulator engine, which is like an interpreter on steroids, and I can't even imagine embracing the idea of emulating the execution of JavaScript with the goal of analyzing its execution to discover anything malicious, anything that it decides is bad. Wow. That's like a heuristic on steroids. Anyway, it analyzes incoming JavaScript code for malicious action before allowing it to execute in browsers or email clients.

Tavis wrote: "Despite being highly privileged and processing untrusted input by design, it is unsandboxed and has poor mitigation coverage. Any vulnerabilities in this process are critical, and easily accessible to remote attackers." He explained that exploitation of the bug was trivial once it was known. All it takes is sending a malicious JavaScript or Windows scripting host file to a user via email, or tricking a user to access a booby-trapped file containing malicious JavaScript code. The faulty Avast engine would then stumble over the file, and the attacker would obtain system-level access with no restrictions. They could then have the ability to install malware on an Avast user's device because of Avast AV being present.

So as I mentioned, a week passed after Tavis notified Avast, and who knows what kind of urgency he put on them. We know that Tavis can be ruthless when it comes to imposing a deadline after which he's going to release the news. And he tends to do that with more urgency, the more urgent the problems are that he finds. He probably gave them only a week to do something. The problem would have been also that just fixing the one problem wouldn't have satisfied him. He talks about it not being sandboxed. Well, sandboxing this was probably a big project.

So anyway, they scampered around for a week. Nothing happened. They were probably hoping to get some sort of a quick fix. Maybe they did, and Tavis said no. We don't know what happened behind the scenes. But the problem is probably more systemic. So whatever the case, after a week they decided to completely disable the JavaScript scanner functionality until it could be properly fixed. So now a significant useful but dangerous - and, you know, Avast users should be, I guess, glad they don't have it because especially now that it's been discovered as being flawed and inadequate, it would put them at much more risk.

So they issued a statement saying: "Last Wednesday, March 4th, Google vulnerability researcher Tavis Ormandy reported a vulnerability to us affecting one of our emulators. The vulnerability could have potentially been abused to carry out remote code execution." Uh-huh.

"On March 9th, he released a tool" - as he did on GitHub - "to greatly simplify vulnerability analysis in the emulator. We have fixed this" - and they didn't put it in quotes. I would have - "have fixed this by disabling the emulator, to ensure that our hundreds of millions of users are protected from any attacks. This won't affect the functionality" - what? "This won't affect the functionality of our AV product, which is based on multiple security layers." On the other hand, it will no longer find malicious JavaScript incoming.

So there's no current timeline for when their patch will be ready. And again, this further reinforces what we have been saying, Leo, recently, that the third-party AV is really becoming a problem. It seems to be also the problem that Microsoft Update is stumbling over time and again.

Leo: Yeah, that's right.

Steve: Remember those cute days of Firesheep?

Leo: The good old days.

Steve: The good old days where it was easy to take over an account. You simply looked for cookies flying by on an unencrypted connection or on an open WiFi and grabbed the

cookie and started sending it yourself. Suddenly you were logged in as the person whose cookie you captured because - and I'll cover this a little bit more briefly in a second - cookies are by their nature static. They're just little blobs which the browser regurgitates to the server, which is an inherent vulnerability that has not yet been solved.

So it turns out that Kaspersky's security announcement was titled "CookieThief: A Cookie-Stealing Trojan for Android." They discovered a bunch of trojans under the banner Trojan-Spy.AndroidOS.Cookiethief. They said: "A combination of new modifications to Android malware code has given rise to Trojans able to steal browser and app cookies from compromised devices."

So on Thursday, March 12th, 12 days ago, the guys at Kaspersky revealed a new malware family which, thanks to their actions of discovery, they dubbed "CookieThief," and the actions that this malware family takes. It uses a combination of exploits to acquire root rights on an Android device, which it then uses to steal the user's Facebook cookie data. And it was interesting that it's Facebook.

Kaspersky wrote: "We recently discovered a new strain of Android malware. The Trojan" - and then they give that same name - "turned out to be quite simple. Its main task was to acquire root rights on the victim device, and transfer cookies used by the browser and the Facebook app to the cybercriminals' server. This abuse technique is possible, not because of a vulnerability in the Facebook app or the browser themselves. Malware could steal cookie files of any website from other apps in the same account and achieve similar results."

So as we know, cookie stealing is all about account hijacking. The only way - and this is the problem, the fundamental problem - in which a user's logged-on session is maintained is with one or more secret cookies which, these days, are marked "secure" so they will only be sent over TLS browser queries. No more simple-minded Firesheep, which simply snagged cookies over the HTTP channel after the user's connection reverted to HTTP following a brief secure interchange of username and password during logon.

But simple static cookies represent a vulnerability. Unlike a more dynamic system, like SQRL, which always requires the client to solve a cryptographic problem and to return its solution with every query, thus reproving its identity every time, cookies are just static blobs of data. Although they might be changed periodically, they're simply regurgitated by the browser, which always returns the most recently set cookies for the domain. And of course the emergence of OAuth, which uses the logged-on status at one site to authenticate the user's identity at other sites, means that someone obtaining a user's Facebook cookies also obtains the ability to impersonate them elsewhere, since they appear to be validly logged onto Facebook. This would allow them to log on as that user at other sites which offer a "Login With Facebook" option, thus spreading the impersonation further.

Kaspersky wrote: "On the command-and-control server we also found a page advertising services for distributing spam on social networks and messengers," they said, "so it was not difficult to guess the motive behind the cookie theft operation." In other words, they've got a trojan which is out there in the wild, obtaining root access for the purpose of obtaining Facebook logon credentials, sending them back to the command-and-control server. So they're building up a library of valid Facebook authentication, and advertising services for distributing spam on social networks. In other words, they're going to accept spam advertising for sale and then turn around and use this continuous churning base of logon credentials to send the spam through the people's Facebook accounts whom they have commandeered.

It turns out it's a little more complicated to do this and get away with it. They're unable to send those out directly because the fact that that would come from a different IP

might set off Facebook. So there's another piece of this which is a proxy which is also installed as part of this malware such that the proxy logs on as the user, using the stolen credentials, accepts the advertising from the command-and-control server, and then sends it from the user's own phone under their credentials. So anyway, this is sort of an interesting anatomy of an example of the way that an Android malware system is able to commercialize people apparently sending spam from their own accounts out onto the Internet. Amazing what these guys will do.

So Pwn2Own Spring 2020.

Leo: Oh, I love this. I love this. It's hysterical.

Steve: Last Friday was the second and final day of, well, the second and final of two days of the 2020 Spring Edition of HackerOne's always interesting Pwn2Own hacking contest. This year's winner is, perhaps unsurprisingly...

Leo: Yes.

Steve: Our listeners will know them well, Team Fluoroacetate, formed by the security pair Amat Cama and Richard Zhu. They won the contest after accumulating nine points across the two-day competition and extended their dominance by winning their fourth tournament in a row. There were, let's see, one, two, three, four, five, six contestants. They were at the top with nine. Then the next one was at seven, and then four, four, three, and zero. So got a nice spread. And interestingly, I mean, I just have to wonder, you know, these guys just seem to pull this stuff out of their hat. And it's like, well, okay.

Leo: You know they save it all year. I mean, they're just - that's why it was a little concerning because CanSecWest in Vancouver, because of COVID, was called off. So I love how they did this. They had the hackers mail in their exploits.

Steve: Right. So of course since no one is currently meeting face to face, and since the Pwn2Own hacking contest has always been held during the CanSecWest cybersecurity conference occurring in the spring in Vancouver, which as you said, eh, this year, this year's Pwn2Own is the first-ever hacking contest hosted virtually. Although it sacrificed some of the in-person drama, the participants sent their exploits to the Pwn2Own organizers in advance. The organizers then ran the code during a live stream with all participants present. So that's still, that's - yeah. And it still preserved much of the breath holding because it was like, okay, is this going to work?

Leo: Plus I think that that's a good, I mean, if your exploit can be automated that way, that's a big deal; right? You don't have to sit there and go, let me try this. Oh, no, that didn't work. Let me try this.

Steve: Yeah.

Leo: If you can make the script, and it works, more power to you. Right?

Steve: Yup. It doesn't require your last-minute sort of finessing to, like, stroke the Enter key just right in order to slip that bug through.

Leo: I think the scores were a little lower this time because of that, to be honest. Because I think there is a certain amount of that.

Steve: Yeah. Well, only one failed. So during the competition's two-day schedule, six teams managed to hack apps and operating systems including Windows, Mac, Ubuntu, Safari, Adobe Reader...

Leo: All of them. All of them.

Steve: ...and Oracle Virtual Box.

Leo: Yeah.

Steve: All the bugs, as always, exploited during the contest were immediately reported to their respective companies. So what happened? The first exploit of the first day was attempted by the Georgia Tech Systems Software and Security Lab team, who targeted Apple Safari. And get this, Leo. With a macOS kernel escalation of privilege, so a browser-based kernel escalation attack. Yikes. But the exploit succeeded when the Georgia Tech team used a six-bug exploit chain.

Leo: Wow. Wow. Wow.

Steve: Which just makes me shake my head. I just think, oh, my god. Chaining together a series of six small flaws in order to amplify the effect of each and get yourself in. They managed to pop up the calculator app on macOS and escalate its access rights to root. So that demonstrates they could run anything that they wanted to, feed it commands to do things behind your back with full root access. And they earned a well-deserved $70,000 U.S. and seven Master of Pwn points for that.

The second exploit attempt was launched by Fluorescence, which is just Richard Zhu by himself.

Leo: Yeah, I thought that was interesting. So he works both as a team and by himself.

Steve: Right.

Leo: He did pretty well just by himself.

Steve: Yes, he did. He targeted Microsoft Windows with a local privilege escalation and succeeded. He used a use-after-free vulnerability in the Windows API to escalate his local privilege. And that is to say, as we know, as I was just saying, that local privilege

escalation, the idea of not running as an admin being protection against some zero-day GDI flaw shouldn't give anyone any comfort because local privilege escalations are all over the place. Anyway, he found another one, earning himself $40,000 and four points toward Master of Pwn.

The third exploit was brought by Manfred Paul of the RedRocket CTF team, who targeted Ubuntu Desktop with a local privilege escalation.

Leo: And how did he get in? Because I want to know about that one.

Steve: And it, too, was successful. He's a newcomer to Pwn2Own who used an improper input validation bug. So there again, a bit of an interpreter.

Leo: Sanitize your inputs, folks.

Steve: You're looking at input. You have to be very careful, at your code, what looks at input because it's possible for the input itself to attack you, which you're trying to prevent from a sanitation standpoint. And there was a problem. He was able to escalate privileges and earn $30,000 and three Master of Pwn points. And in all of these cases we're not getting full detail.

Leo: Yeah, because I'd like to know what app it was or, you know, what was it that let them do that?

Steve: Well, we've got to give the companies behind these things time to go, ooh, crap, and then push out a fix before it comes more widespread.

Leo: Is it a widespread kernel flaw, or something unique that Ubuntu was doing, or what?

Steve: Yeah. And we're just happy these guys are wearing white hats.

Leo: Well, that's one point to make is that they have to turn these exploits over to the company afterwards.

Steve: Right, right. Yeah, well, this is a HackerOne-run operation. So everything is about the good side, where Zerodium is the bad side of this whole deal.

So for the fourth exploit, the Fluoroacetate team was back, setting their sights on Microsoft Windows with a local privilege escalation. As I said, they're everywhere. They succeeded in leveraging a use-after-free bug in Windows.

Leo: Keep hearing that, don't you.

Steve: Yeah. To escalate themselves to full system privilege and earn $40,000 and four more Master of Pwn points. The second day began when Star Labs targeted Oracle's Virtual Box for the first exploit in the virtualization category, and it succeeded. The researcher used an out-of-bounds read bug for an information leak and an uninitialized variable for code execution on the Virtual Box hypervisor.

Leo: That's a little more sophisticated.

Steve: So that is a VM escape. Yeah, you don't want code running on the hypervisor. He did it. And he took home $40,000 for his effort, and got four Master of Pwn points.

The Fluoroacetate team came back one final time for the sixth exploit of the competition, targeting Adobe Reader with a Windows...

Leo: They shouldn't get any points for that. That should be just, oh, yeah, I know, okay.

Steve: Exactly what I thought. What I thought, for sure, oh, come on. Except it turns out it was a complex combination of Reader and obtaining a local privilege escalation. They employed a pair of two use-after-free bugs, one in Acrobat and one in the Windows kernel, in order to hike their privileges and to gain control, complete control over the system. That one got them $50,000 and five points toward the Master of Pwn.

Only one exploit failed during the competition, which was the seventh and final one. It was made by the Synacktiv team. They targeted VMware Workstation, obviously also in the Virtualization category. But as I noted, the attempt failed when they were unable to get their exploit to function in the allotted time. So overall, despite doing this virtually, we were still able to have an interesting and worthwhile and hopefully motivational Pwn2Own meeting. These guys are taking home some serious bread, and it's definitely worth...

Leo: I think they should stream this like eSports. I mean, seriously, this is cool.

Steve: It would be, really, yeah.

Leo: Yeah, yeah.

Steve: It seriously is. So I had a couple of just sort of observations and discoveries relative to where we are today with the coronavirus that I wanted to share with our listeners. I wanted to mention testing briefly because we've all been hearing about testing. What has become clear to me as I've been reading a lot about this is that this PCR test, technically called an RT-PCR, Reverse Transcription Polymerase Chain Reaction, it had the benefit of being quick to design, but the serious liability of never really being scalable.

And frankly, after I've understood what's involved, I'm astonished that other countries are managing to perform this test. As far as I know, this is what they're still using. They're able to perform this test in the kind of volume they are. It is extremely labor intensive. It's very slow. It takes at least four hours, and often more. And it's expensive.

It ties up equipment during that time. It tends to burn through the very scarce personal protective equipment because when you get this flexible synthetic - it can't be a cotton swab. It has to be a synthetic swab because cotton interacts with the virus.

When you get this flexible synthetic swab shoved further up your nose than you knew your nose went, it's very uncomfortable. People invariably cough or sneeze, thus polluting the local environment. And that means that the person administering the test has to change and discard this round of personal protective equipment. It's just never going to be a solution for mass testing.

And it's turning out that there was some interesting input back from China that suggested that they're seeing a significant level of false negatives, meaning it is not successfully detecting somebody with the problem. The reason is that the actual virus is operating down in our lungs, and at the very top of our upper respiratory system there just may not be any. The virus apparently takes up shop for three or four days up in our throat, which is why a sore throat is the very first symptom that most people report. And it is often transient, and then the virus moves down into our lungs, where it of course causes famous all kinds of havoc.

Anyway, where we're headed, and we're there in terms of the design, but not yet in terms of availability, there's a broad class of tests known as ELISA tests that came online in 1971. ELISA is an abbreviation for Enzyme-Linked ImmunoSorbent Assay. And here's a description from one of the very many pieces of work that's currently underway. The guys that have designed a test, just to give you a sense for it, said: "To create the test, the researchers began by designing a slightly altered version of the 'spike' protein on SARS-CoV-2 outer coat."

They said: "The alterations made the protein more stable for use in the lab. That protein helps the virus enter cells, and it is a key target in the immune reaction against the virus, as the body churns out antibodies that recognize the protein and tag the virus for destruction." So in other words, this is the thing that they want, that our body needs to learn to recognize, and that they want their test to be able to use.

They said: "They also isolated the short piece of the spike chain protein called the receptor-binding protein (RBD), which the virus uses to attach to cells it tries to invade. They then used cell lines to reproduce large quantities of the altered spike proteins and RBDs. Those lab-made molecules provided the basis for an ELISA test, in which antibodies in a sample of blood or plasma trigger a color change when they recognize a target protein - here, an RBD or the spike protein. Initial tests of four blood samples from three confirmed COVID-19 patients, and from 59 serum samples banked before the start of the outbreak, showed that the test worked, as antibodies to SARS-CoV-2 bound to the test's proteins. It showed positive results only for the COVID-19 patients and not for any of those controls."

So that's a little bit of sort of behind-the-scenes of how these tests are developed, and the fact that we're in the process of moving past this existing PCR test with all of its many problems. It's going to take a while for labs to ramp up, for the supply chain to get into place, and for these to get standardized. The FDA has already approved one of these.

And as we know, one of the great things in my opinion that our administration has done is to really take the shackles off of the typical way overprotective FDA safeguards in order to get these things out into use as soon as, I mean, maybe mistakes will be made. But what we really need is visibility. And having a finger prick test that anyone can do in a few minutes will be incredibly useful. It'll give us the first sense of visibility.

And the other thing I just sort of wanted to mention from a math standpoint is how I'm just infuriated every time I see the popular press say there are 30,000 cases of COVID-19 in the U.S. No. There are 30,000 people who were tested positive. But as we know, globally, not just in the U.S., but a tiny fraction of people were tested. And so we still have absolutely no sense for how widespread this is. And this is no one's fault. This PCR test, as I said, is a disaster, with multiple reagents. It's extremely labor intensive to do. It was never scalable on a scale that we need.

But while I've been thinking about this, I've realized that we are never going to test all, what is it, 330 million citizens in the U.S.? Maybe we'll test a statistically significant sample, so that from a statistics standpoint we can know with some level of certainty that, okay, we found this percentage of positives in a heterogeneous sample cohort of a certain size. So we know that with reasonable certainty it can't be more than this or fewer than that. But still, we don't have 100% visibility.

What occurred to me, and it's a little bit morbid, but it's true, is one thing that we absolutely know is the death toll that this is taking. That is, we know the other endpoint of some percentage of these sicknesses. And so the only thing, because how many people "have it," quote unquote, is just complete nonsense. It's not of any use. What is of use is unfortunately the shape of the death curve.

The problem is it lags, as we know, at least on the order of two or three weeks behind the "who has it and could be tested to have it" curve. So that means that, as we know, everything is moving very rapidly here. So unfortunately, using the count of people who have died isn't useful for making real-time policy decisions which you'd like to be making three weeks earlier, but we just still don't have that data.

But for me, from an epidemiological standpoint, as long as the curve is going upwards, we don't know where it will end. It is only when the slope starts to slow, or hopefully reverse, and that we stop seeing an increased rate of dying, that we'll know that three weeks earlier than that things were being done that were slowing down the spread. So anyway, I just sort of wanted to - I've talked about this a few times in other forums, and I sort of thought it's interesting to remind ourselves that the one real number we have, sad as it is, is how many people this thing is taking out. And nobody needs tests to determine that, sadly.

Also, last week I shared a GRC shortcut, grc.sc/covid, which was that Ars Technica backgrounder. I have two more. The first one is, oh, my god, it is so wonderful. It is an animated presentation of sort of the whole situation. It's an explainer for the whole family. And it's so well done, so kind of with a little bit of a tongue in cheek, not taking itself very seriously. I hope that our listeners will grab it and arrange to put it on a big screen for the whole family to watch. And that's grc.sc/covid2. It basically walks its viewer through this entire thing, like why exactly does hand-washing work.

Leo: Yeah. I loved the info about how the virus works that gives you the information about cleaning up. This is from a - 16 million views on this, by the way. It's from a channel called Kurzgesagt, which means, I guess, in a nutshell.

Steve: Mark Thompson knew about it, and he said, oh, yeah, I love the...

Leo: Yeah, they do a lot of good science stuff. And I'm guessing he's German because of the accent, but also because of the name. But it's very well done, yeah.

Steve: Anyway, I cannot recommend it highly enough to our listeners. You can find it, grc.sc/covid2.

Leo: Yeah.

Steve: Okay, now, the third one is way higher level. This is not for everybody. Certainly not, well, so it is medical school-level whiteboard explanation of where the coronaviruses have been originating, the original source of the pathogen, its mutation over time. It's a whiteboard presentation by a guy who I grew increasingly respectful for.

Leo: Ninja nerd science.

Steve: Wow. It is really good.

Leo: He's got good whiteboard skills, I can tell.

Steve: Yes, he does. And he's, like, rubbing mistakes out. He's changing colors. He prints clearly enough that you're able to see what's going on. Anyway, so for our listeners who can weather, like, the real down-in-the-details medical science, I mean - and it's funny, too, because one of the guys in the grc.health newsgroup named Ian, after he watched it he posted, "Okay, I'm ready to move on to the anxiety counseling video now."

Leo: Yeah. Because it is scary, yeah.

Steve: It's grim.

Leo: Yeah.

Steve: It is grim. But anyway, so grc.sc/covid2 for the family-friendly amazing animated - and this is the one you're going to want to share with everybody that you guys know. And then covid3 for, like, okay, just sit down, and you'll be in medical school for about 50 minutes. You'll come out the other end having a real sense for what's going on.

Leo: I will watch that one tonight because I watched the other one, and I loved it. And in fact we did exactly what you suggested, which is put it on the big screen and got everybody to sit down and watch it because it was a really good explainer, the In a Nutshell.

Steve: Oh, my god. It's just amazing.

Leo: Yeah.

Steve: Okay.

Leo: Can I ask, before we get to the topic of the day...

Steve: Oh, yeah, yeah, yeah.

Leo: A listener has called me now twice on the radio show to say what the hell size of the mug is that that Steve - no. Steve and his giant mugs, of course, occasions interest among all. But he says at one point on a show you mentioned why you don't want to use wireless headphones, like there might be some health risk associated with that.

Steve: Oh, gosh. I have avoided talking about this.

Leo: Okay.

Steve: We actually do have some science about this.

Leo: Okay.

Steve: There is something in the structure of our brain known as the "blood brain barrier."

Leo: Yes.

Steve: It sounds like some Wall of Jericho or something, but it's not. It's the fancy name for a different form of endothelial lining and all the capillaries which feed nutrition to our brain. So it's just a different interface lining for blood-borne stuff, allowing only some things to get through. And for me, as someone who's like wanting to experiment with things, interesting chemicals like GABA are unable to pass the blood brain barrier. So you have to use a precursor, tryptophan, which is able to pass, being a simpler amino acid, in order to get tryptophan in, and then you get an increased production of GABA inside. And, you know, and so forth.

So what has been shown is that cellular frequency and power significantly increases the porosity, the porousness of our blood brain barrier. Some researchers were able to place some chemicals outside which would normally never be seen inside, and then use nothing more than the equivalent power of a cell phone transmitting at your head and demonstrate uptake into the interior of our brain these chemicals.

Leo: So in other words, it might make the blood brain barrier more porous. It's funny because this is, you know, we talk about the damages from RF. What I consistently say is, well, the problem with that of course is that RF is not the kind of radiation that would damage cell structure. And it diminishes with distance by such a great deal, and it's such low power to begin with that it seems unlikely, and there's never been any evidence that cell-style radio transmissions or for that matter

Bluetooth would cause damage to the cells. It's not ionizing radiation. But this is a completely different kind of damage. And I doubt anybody's even looked at this except for this study that you're quoting.

Steve: It's why I believe it is science-based. And I'll see if I can find the article.

Leo: And this is not - this is something caused by the RF, not by, for instance, one of the things I've heard some people say is, well, if you're worried about cell phone radiation, and you use Bluetooth headset to protect yourself, or you don't want to use Bluetooth because of the radiation there, plugging in a wire just puts a direct line from the cell phone into your ear. So it wouldn't necessarily be...

Steve: Yeah, it does serve as an antenna.

Leo: Yeah.

Steve: Yeah. You know, Lorrie very, very reliably holds the phone about a foot away. It is the case that the power drop is dramatic.

Leo: Yeah, it's the inverse of the square of the distance.

Steve: It's also the case that, when it's at your ear, you have a transmitter there that is reaching out to a cell tower. So, and we know how quickly it drains the battery in our phone. I mean, it's not a low bit of power. We're all wanting the convenience of this not being a problem. I just hope it won't be a problem.

Leo: There's no evidence of increased brain cancer from cell phone use. And we've had it now for more than a decade. But this interesting - this porousness of the blood brain barrier is intriguing. I don't know what the consequences would be. It might be illnesses, not cancer illnesses, but other illnesses or other issues.

Steve: Well, and who knows; you know? Alzheimer's or dementia or other things that seem to be on the rise that we're like, oh, look, why do we have more of this now? Or ADHD that seems to be a plague. You know, it's like we are - we're mucking with something that wasn't designed to happen. So, yeah, fortunately I'm just not a phone talker. I just there's - I never have occasion to do it. And Lorrie's really good about using her speakerphone function.

Leo: Yeah.

Steve: So that's the answer. And I'll find the paper.

Leo: Thank you.

Steve: I imagine our listeners would - and I imagine now that I've opened the Pandora's Box to this...

Leo: You're going to have to put a link in.

Steve: ...a lot of people are going to, "Hey, Gibson, blah blah blah." It's like, okay, slow down.

Leo: Yes. Yeah, want to hear more about that.

Steve: All I'm doing is telling you what I read. And it looked legit.

Okay. TRRespass, T-R-R-e-s-p-a-s-s. The short version is the fixes for Rowhammer have not worked.

Leo: Oh.

Steve: Yeah. We began covering Rowhammer half a podcast, half this podcast ago, six years, in 2014. And remember, we should not confuse Rowhammer with the more theoretical processor architecture attacks such as Spectre and Meltdown. They were important because they showed how the tricky designs used to increase processor performance could be leveraged against us. But they were never easy to pull off in the field. The researchers have shown over and over again, and I'm going to have a little - I found a bullet-point list of things they did. They'd shown that Rowhammer is a much more real and significant threat.

So to quickly recap, researchers six years ago discovered that the main bulk volatile DRAM lying at the heart of every system we use was not nearly as robust anymore as we'd always assumed. Over the years, under the pressure to deliver ever more RAM density, DRAM density, the memory storage cells had been successively reduced in size to the point where they were operating right on the hairy edge of "We can't make them any smaller." And think about it, Leo. Remember the RAM we used to have that was like, a few megs or maybe a gig? Now you're plugging in 64GB in the same size chunk.

So it's like, okay, what's the story? And DRAM parity checking and error correction technologies, which were originally intended to protect against the stray cosmic ray hitting and flipping a bit, they're increasingly being used to buffer the DRAM's underlying reduced reliability. We can think of this in terms of a noise margin where there's a given certainty that a DRAM cell's voltage represents a zero or a one. And over time, in the pursuit of DRAM density, the noise margin has been successively reduced.

Well, the ever-clever ever-loving researchers, and this is Herbert Bos and his gang back at University of Amsterdam and some others, showed the world back then that by forcing atypical DRAM access patterns to deliberately create higher environmental noise, that is, like noise in the area, it was possible to cause modern DRAM to malfunction in such a way that with some control over this, individual bits could be flipped.

Since today's processors use DRAM-based tables to manage their memory virtualization, they then demonstrated, the researchers, all of the various sorts of mischief that could be created by flipping bits in these DRAM-based management tables and much more.

Malicious row hammering processes could give themselves full access to other processes, give themselves read-write access into the Windows OS or any OS kernel and more.

So through the ensuing years, and here's my bullet-list, researchers showed how a Rowhammer attack could alter data stored in DDR3 and DDR4 memory. They showed how a Rowhammer attack could be carried out via JavaScript, via the web, and not requiring access to a PC physically or via local malware. They demoed a Rowhammer attack that took over Windows computers through the Microsoft Edge browser. They demoed a Rowhammer attack that took over Linux-based virtual machines installed in cloud hosting environments. They used a Rowhammer attack to get root permissions on Android smartphones.

They bypassed Rowhammer protections put in place after the disclosure of the first attacks. They showed how an attacker could improve the efficiency of a Rowhammer attack by relying on local GPU cards. They developed a technique to launch Rowhammer attacks via network packets. They developed a Rowhammer attack that targets an Android memory subsystem called ION, and which broke the isolation between the OS and local apps, allowing data theft and total device control.

They developed a Rowhammer attack named ECCploit that works even against modern RAM cards that use error-correcting code. They discovered RAMBleed, a Rowhammer attack variation that can exfiltrate data from attacked systems, not just alter it. And they developed a technique to speed up Rowhammer attacks with the help of field-programmable gate array cards in an attack named Jackhammer. So these guys have been busy.

The solution to all of this Rowhammer mess was supposed to be a series of mitigations collectively referred to as Target Row Refresh (TRR). And thus the T-R-R-e-s-p-a-s-s, TRRespass, Target Row Refresh. And we've talked about that, too, the idea being that noise immunity can be maintained if DRAM rows in the areas of unusually high activity, which we've learned would tend to soften their stored bits, are proactively brought up for refresh more often, thus reasserting their zero-ness and one-ness. TRR has been gradually implemented and has been rolling out over the past six years.

And back then we were talking about, oh, DDR4, that'll be the answer because that's coming online, and that's going to be the solution. They were the first ones to receive the TRR, the Target Row Refresh protections. And for a while vendors believed that they had finally plugged the Rowhammer issue. But you always have to worry when you hear the term "mitigation," as in, well, we really didn't fix it, but we made it much better. Uh-huh. The original Rowhammer guys, led by Herbert Bos and his team from Amsterdam University, in league with researchers from ETH Zurich and Qualcomm, recently released their paper titled: "TRRespass" - spelled their funny way - "Exploiting the Many Sides of Target Row Refresh." I think I'm about to sneeze. Maybe not. I love sneezing.

Leo: Oh, it's so good.

Steve: Nose tickle. In this research - I think it passed - they outlined their development of a generic tool called TRRespass that can be used to upgrade the old Rowhammer attacks to work on the new and improved TRR-protected DDR4 RAM.

So here's how they summarized their work. They have three paragraphs: "After a plethora of high-profile Rowhammer attacks, CPU and DRAM manufacturers scrambled to deliver what was meant to be the definitive hardware solution against the Rowhammer problem: Target Row Refresh. A common belief among practitioners is that, on the latest generation of DDR4 systems that are protected by TRR, Rowhammer is no longer an

issue in practice. However, in reality, very little is known about TRR. How does it work? How is it deployed? And is it actually effective against Rowhammer?" We want to know.

"In this paper, we demystify the inner workings of TRR and debunk its security guarantees. We show that what is advertised as a single mitigation is actually a series of different solutions coalesced under the umbrella term "Target Row Refresh." We inspect and disclose, via a deep analysis, different existing TRR solutions, and demonstrate that modern implementations operate entirely inside DRAM chips."

What they meant by that was that it wasn't clear whether some of this might be done by the DRAM controller, which is upstream of the chips. What they learned was DRAM controllers are still asleep. They didn't bother with this at all. All the responsibility was given to the DRAM chips themselves.

They said: "Despite the difficulties of analyzing in-DRAM mitigations, we describe novel techniques for gaining insights into the operation of these mitigations. These insights allow us to build TRRespass, a scalable black box Rowhammer fuzzer that we evaluate on 42 recent DDR4 DIMMs. TRRespass shows that even the latest generation DDR4 systems with in-DRAM TRR, immune to all known Rowhammer attacks, are often still vulnerable to new TRR-aware variants of Rowhammer that we have developed. In particular, TRRespass finds that, on present-day DDR4 modules, Rowhammer is still possible when many aggressor rows are used," they said, "even 19 in some cases." It actually turns out four is enough. Aggressor rows being the neighborly adjoining rows which are used to induce bit flips.

They said: "...in a configuration we generally refer to as 'many-sided Rowhammer.' Overall, our analysis shows that 13 out of the 42 DIMMs from all three major DRAM manufacturers - Samsung, Micron and Hynix - are vulnerable to our TRR-aware Rowhammer attack patterns, and thus one can still mount existing state-of-the-art Rowhammer attacks. In addition to DDR4, we also experiment with LPDDR4 chips and show that they are susceptible to Rowhammer bit flips, too. Our results provide concrete evidence that the pursuit of better mitigations must continue."

So what has essentially happened is that DRAM manufacturers looked at the existing Rowhammer attacks. They didn't actually solve the Rowhammer problem because, frankly, they can't. The Rowhammer problem does not arise from some defect in DRAM. It is insidious because it arises from the underlying technology of DRAM for which there can be no quick fix.

So what did they do? They designed, the manufacturers, the DRAM manufacturers designed internal secret proprietary hardware-mitigation workarounds for the various specific known Rowhammer attacks. So the various things that were done before, indeed, no longer work. The undaunted researchers reverse engineered what was going on inside the DRAM controllers by very carefully examining the updated DRAM. They used a side-channel analysis of DRAM in order to figure out what the algorithms were in the DRAM controllers. And once they'd learned the tricks that had been incorporated, they simply evolved different attacks to bypass those new tricks.

They had three observations in their notes. They said the TRR mitigation carries out a targeted refresh on every refresh command. Second observation, the mitigation can sample more than one aggressor per refresh interval. The third observation, the mitigation can refresh only a single victim within a refresh operation.

So in other words, those are examples of the things they learned about what was being done in DRAM controllers. And they came up with: "Based on these observations, we conclude that hammering more than four rows should circumvent the mitigation. We

confirm this by running a test on our FPGA infrastructure with standard conditions." And I'm not going to go through their conclusions because they're largely repetitious.

But I did highlight in red here: "Our results provide evidence that the pursuit of effective Rowhammer mitigation must continue and that the security by obscurity strategy of DRAM vendors puts computing systems at risk for extended periods of time." And I missed the thing that I thought I'd highlighted there. I know what it was. Oh, here it is.

"This paper shows that, despite significant mitigation efforts, modern DDR4 systems are still vulnerable to the Rowhammer vulnerability, and even more vulnerable than before, once the mitigations are bypassed." So essentially the DDR4 memory that again bumped up its capacity and increased the fundamental problem implemented some heuristic design tricks to solve, they thought, the increased fundamental problem resulting from reduced noise margins in DDR4 higher speed, higher density chips.

So they tried to work around that by making a fancy controller. But it turns out that's algorithmic. You can figure out the algorithms, and you can sidestep them. So, you know, tip of the hat to the researchers who did this. Without this, we would all just say, oh, look, Rowhammer no longer works. Yeah. It's worse than it was before because now we have DDR4, and the protection algorithms can be bypassed.

Leo: Wow. What a story. Steve, thank you. As always, Steve Gibson, ladies and gentlemen. Let's hear it for Steve. Round of applause for Steve Gibson, man of the hour. Security Now! is every Tuesday, 1:30 Pacific, 4:30 Eastern, if you want to watch us do it live. That's 20:30 UTC. You can just watch the live stream at TWiT.tv/live. You can listen to live audio, too.

I think more and more, you know, because of COVID-19, conferences and events are disappearing. People are not getting together in person. But a show like this is a great opportunity to get the information you need without having to leave your house. So we really appreciate it. In fact, if you subscribe, you'll get every episode. You don't want to miss an episode. Just go to your favorite podcast client and subscribe.

Steve has 16Kb versions, for people who don't want to use the bandwidth, on his website, GRC.com. He also has transcriptions written by humans, not by machines. So Elaine does a great job writing those out, and they make it a little easier if you want to follow along as you listen. That's all at GRC.com. While you're there, get a copy of SpinRite, the world's best hard drive maintenance and recovery utility. Steve's scribbling right now, writing like crazy to make the next version of SpinRite. You'll get it for free, you'll even be able to beta test it, if you buy SpinRite right now: GRC.com. Lots of other stuff at GRC.com, including...

Steve: Actually, this morning I posted a "Thanks for your patience" note to the grc.spinrite.dev group, who have been patiently waiting for me to return. I was down for the week...

Leo: You were sick.

Steve: ...between these podcasts, yeah. I was...

Leo: I'm really glad you rested and took it easy. And I want to agree with Lorrie, don't do it if it's going to take it out of you.

Steve: Well, I really thank the global pandemic for fitting nicely in between Security Now! episodes because I was able to emerge for a little burst of podcast last week, and then I was pretty much laid flat for a few days.

Leo: You need to take it easy, Steve. We can't afford to lose you.

Steve: I love my body. It always performs for me.

Leo: Lisa wanted me to tell you that you've got to take care of yourself because in this COVID day and age we get more and more advertisers who want to be on Security Now! because they know this is how you reach that audience. When you can't have a conference or you can't get in front of them any other way, advertising with Security Now! is the best way to do it. And they always mention you and how much they love you.

So thank you, Steve. We really appreciate it. You may be the last man standing as the TWiT Network slowly dwindles down to nothing. But this show, this show will go on. We have, as I mentioned, at TWiT.tv/sn we've got audio and video of the show. But again, the best thing to do is subscribe in your favorite podcast application so you don't miss a minute. And we will be back here next Tuesday. Steve will be fit as a fiddle, and we'll have a lot more information for you. Thanks, Steve.

Steve: Yes, sir, my friend. Glad to be here. Glad to do it. And I'll see you next week. Bye.