SECURITY NOW!

Transcript of Episode #705

Spoiler

Description: This week we look at the zero-day exploit bidding war that's underway, the NSA's release of Ghidra, Firefox's addition of privacy enhancements which were first developed for the Tor version of Firefox, a pair of zero-days that were biting people in the wild, news of a worrisome breach at Citrix, the risk of claiming to be an unhackable aftermarket car alarm, a new and interesting "windows developers chatting with users" idea at Microsoft, a semi-solution to Windows updates crashing systems, detailed news of the Marriott/Starwood breach, a bit of miscellany from Elaine, a SpinRite question answered, and then we finish with SPOILER - the latest research exploiting yet another new and different consequence of speculation on Intel machines.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-705.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-705-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here with some interesting topics, including why would somebody pay half a million dollars for a hypervisor bug, a step-by-step discussion of how the Marriott hack happened, and a new feature in Firefox to prevent letterboxing. What? Yes, it's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 705, recorded Tuesday, March 12, 2019: Spoiler.

It's time for Security Now!, the show where we cover your security, your privacy, talk a little bit about how the Internet works, how computers work with this guy right here who knows about everything, understands it all, and even is able to communicate it, a rare combination of abilities, Mr. Steve Gibson. Hello, Steve.

Steve Gibson: Actually, you know, I was seriously thinking about getting a T-shirt printed: The more I learn, the less I know.

Leo: Yeah, well, that's the truth, isn't it.

Steve: Because that really is the case. The arrogance of youth who sort of think, oh, I've got this all figured out. And then when you really start digging in you think, whoa, this is a lot bigger than I thought it was.

Leo: Well, see, that's because you're intelligent. You're not subject to the Dunning-Kruger effect.

Steve: What?

Leo: The Dunning-Kruger effect is a well-known problem of cognitive bias in psychology in which people of low ability have mistakenly assumed that they're brilliant; they're smart.

Steve: Oh, well, yeah. And after all, really, how would you know?

Leo: Right. And it turns out the dumber you are, the smarter you think you are. And the corollary would be the smarter you are, the more you realize I don't know anything.

Steve: I have no clue about what's going on.

Leo: And that's kind of, I think, a form of wisdom, really. As you get older, you realize, yeah, no.

Steve: Well, and I've always said that one of my favorite phrases is "I don't know." I mean, it really came to light in the early days. I was working with an engineer who worked for me on v3 of SpinRite. And this is the first time I sort of had an engineering group, like other people to test things and to interact with. And his name was Jim. A neat guy. He was a GATE child, so I'm sure his mother thought he was just brilliant.

Leo: Gifted and Talented, or something like that, Education.

Steve: Yeah, GATE, exactly. And so he kept insisting, when there was, like, something in SpinRite didn't work right, he insisted on guessing. He would just jump ahead and just guess what it was. Which is not my approach. And because I was working with him, the contrast of our approaches was very clear to me. And I said, "Well, Jim, maybe. But now we have a problem." And he said, "What?" And I said, "Well, you have a stake in the outcome now."

Leo: Ah, yes.

Steve: "So we all have egos. We're all human. Now you're going to work to prove yourself right."

Leo: That's right.

Steve: My interest is to find the problem, and I don't care. I mean, I'm not leaping ahead. I don't need to guess and then plant a flag on it. I just want to go get rid of it.

And it happened over and over and over. And I'm not sure I ever worked him out of that problem. But it did, it highlighted for me, as these things often do, that my nature is to sort of say, hmm, okay, let's go find what's wrong. And also I would say to him, "Well, that might be. But it might not be." And the point being that the beginning of the search for knowledge is to say I don't have any.

Leo: Yeah, I don't know.

Steve: I don't know. And then that's where you want to start.

Leo: We should teach kids that in school, that it's okay not to know; that it's good not to know.

Steve: Yeah, well. It's probably just genetic and hormonal, and at this point...

Leo: Yeah, it probably is. As we know, none of this is rational; right.

Steve: And you also sort of have to be able to afford not to know.

Leo: Yes. That's a luxury, too.

Steve: And there is a - exactly, exactly. So this is Episode 705 for March 12th. This is Patch Tuesday. And as always happens, I'm so, like, every second for the last eight hours, four hours this morning, four hours yesterday, has been in assembling this podcast for our listeners, that I ran out of time, didn't have any chance to go digging.

So next week I will - in fact, there's one real question because there's a zero-day in a 32-bit version of Windows 7 which is being exploited in the wild, or was being when it was being leveraged by another zero-day in Chrome, which Google immediately killed last week. But I'm curious to know whether Microsoft also killed it in the 32-bit version of Windows 7, and I don't know. But I'll find out.

"Spoiler" is the title of this week's podcast, and it's supposed to be in all caps. But because it's like kind of an abbreviation, except the research paper titled "SPOILER" in all caps, then says, colon, and then this name, somewhere I read that they just got - they were so tired by the time they got through with the research they didn't have the oomph left to reverse engineer a name for the paper that created Spoiler as an abbreviation. So they just said, okay, we're going to call it Spoiler, and it doesn't stand for anything. But it is interesting. We're going to wrap up the podcast as we typically do by talking about that in some depth.

But we're first going to look at a zero-day exploit bidding war that is now underway. We have the NSA's release last week of Ghidra that we predicted. That was at the RSA conference that the NSA formally released Ghidra, that is, their interactive reverse engineering tool. So we'll talk about that. We have Firefox's addition of privacy enhancements which were first developed for the Tor version of Firefox. This, as I was mentioning, a pair of zero-days that have been biting people in the wild, one in Win7 (32) and the other combined in Google's Chrome browser.

News of a worrisome breach at Citrix, and sort of their CSIO really underplayed it, I think, compared to some later news that came out. The risk of claiming to be an unhackable aftermarket car alarm. You never really want to say "We're unhackable." It's like, oh, that's just dangling the bait in front of the penetration testers. And oh, boy, did we find out otherwise.

We have a new and interesting "Windows developers chatting with end users" idea from Microsoft. A semi-solution to Windows Updates crashing systems. Detailed news finally of the Marriott/Starwood breach. A bit of miscellany from Elaine. A SpinRite question answered. And then we're going to finish with Spoiler, the latest research exploiting, get a load of this, here we are a year downstream, yet another new and different consequence of speculation on Intel machines.

Leo: Oh, no. Oh, no.

Steve: Yes. It turns out there's another thing Intel did to get more performance that is speculative, and it allows a dramatic leveraging of Rowhammer.

Leo: Hmm.

Steve: So, yeah, so it's sort of the - and that's part of the title of their paper that we'll be getting to. And we have, nobody knows yet, the Picture of the Week.

Leo: What are you laughing at?

Steve: Now I'm wishing that I had titled this podcast "Vague, but Exciting."

Leo: Okay. Okay.

Steve: Because that was the comment that Tim Berners-Lee's boss wrote at the top of his document which proposed the creation of something which grew into, 30 years ago...

Leo: That's hysterical.

Steve: ...the World Wide Web.

Leo: Oh, my god. Vague, but exciting.

Steve: Vague, but exciting. So our Picture of the Week is a picture of the document. And you can see in the upper right-hand corner it says March 1989. So that was 30 years ago. And what is not clear, and I'm not sure how much time Tim's boss spent staring at this diagram, but this is a self-referential diagram that is a diagram about this document because you can see where, in the lower right it says "CERN, DD division, OC group, RA section, Tim Berners-Lee." So there's a hierarchy. And there's an arrow, "wrote This document," which then describes - it's pointing to "Hypertext," which includes

"Hypermedia" pointing down, and "includes Linked information for example." And anyway, so it's...

Leo: Hyper Card's actually in this document, which is cool.

Steve: Yeah, yeah. And in fact there was some commentary that suggested that Hyper Card, had it been network-aware, which it wasn't, could have swayed things in a very different direction because, I mean, Hyper Card might have had a very different future had it been network-aware. So anyway, this thing is under the heading "Information Management: A Proposal." The abstract - and again, I don't disagree with the boss who said "Vague."

Leo: Vague, but exciting.

Steve: Yeah. The abstract says: "This proposal concerns the management of general information about accelerators." And we're talking about linear or circular, I mean, like atomic accelerators. I can't think of the right word.

Leo: Particle accelerators.

Steve: Particle, thank you, particle accelerators, "...and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system." That's all. I mean, so there's the abstract. And so you can imagine the boss saying, "Huh? That's kind of vague, but I'm excited." Anyway, so the point is that it was 30 years ago that Tim wrote this letter or produced this proposal which foretold - and actually, I mean, from that, over the course of sort of some evolution. I have a link to it below the picture, if anyone is interested. Anyway, I just wanted to commemorate this...

Leo: Yeah, it's a big deal.

Steve: ...this 30-year-ago history. And so begat what we now know as the World Wide Web. And I'm thankful. I mean, it's interesting. He talks about the loss of information. And we do have, although there is some link fade where you go after something where the domain got lost and something that we were hoping to find is gone, although we do have the web archive. And I found myself several times actually in the past week using the web archive to find some information, which it had grabbed following links and archiving of domains, that had since been disappeared. So it is nice to have that. But it also is the case that, boy, just anything you want to know. I just make so much use of the Internet for, you know...

Leo: Oh, my, yes.

Steve: ...expanding my knowledge. Okay. So last week's topic, as we know, was "Careers in Bug Hunting." While we were delivering that podcast last Tuesday, and although we were primarily talking about HackerOne, I mentioned Zerodium as an admitted alternative cash-out source if someone found a particularly tasty and important zero-day flaw, "important" meaning in something where it really matters, like in a named mainstream product, that would allow somebody exploiting the flaw, we won't characterize them, to gain some information that they're trying to get.

As it happens, while we were delivering that podcast, Zerodium was upping the ante. They tweeted: "We're paying up to $500,000 for zero-day exploits targeting VMware ESXi," and they said in parens "(vSphere) or Microsoft Hyper-V, and allowing Guest-to-Host escapes." In other words, VM escapes to get to the underlying OS. They said: "The exploits must work with default configs, be reliable, and lead to full access to the host. Contact us," and then there was a link in the tweet. And they signed off "Zerodium (@Zerodium) March 5, 2019." Half a million dollars for something that someone discovers in Hyper-V or in VMware.

So it turns out Zerodium does, and has in the past, done what they call "acquisition raids," offering limited time increased incentives for classes of flaws they have a particular interest in obtaining, either I presume to balance out their exploit portfolio - and of course, as we know, these flaws are being found and patched on an ongoing basis. So there is some churn in their portfolio.

And, boy, I would love to be on the inside of the contract that they have with a purchaser of one of these to get a sense just for what it looks like. Clearly they're paying a lot for it, and they're not wanting it to be disclosed. It is in the interest of the purchaser of the flaw who wants to exploit it not to disclose it so that they can use it against presumably in targeted attacks.

This is nothing you're going to spray all over the Internet because it'll be found immediately by all of the different security services that are looking for anything like this. So, and presumably the purchaser, if Zerodium is paying half a million, we don't know how large the other side market is, you know, how many people are purchasing these. But they're going to be paying a pretty penny. We do know that it's typically governments and intelligent services of governments that are paying.

Leo: So that's how this economically works with both Zerodium and HackerOne is that they're a marketplace. They're reselling it on to somebody else.

Steve: No. HackerOne is not a reseller.

Leo: Zerodium is.

Steve: Yes. HackerOne, for example, would be contracted by GE to be like the middleman to handle all the mechanics and payment and terms and conditions. So the idea would be, GE being a responsible provider of software recognizes that the best way to get their bugs found is to offer a bounty. So there, HackerOne is a contracted middleman. But Zerodium is very different. Zerodium, you know, they're selling them to the dark side, arguably.

Leo: Got it. Got it.

Steve: Anyway, so they do these periodic what they call "acquisition raids" to increase the incentive, either to balance out their portfolio, maybe as I said some of the things in their portfolio are killed because they're patched, or perhaps they have a customer who is making a particular request. And so, again, they're standing between the security side, us, the public, and the back side. And we don't know who is, you know, what the terms and conditions or anything about the interaction on the other side of this.

So maybe they have a customer who says, okay, we really have a need for a Hyper-V zero-day. So we're willing to give you a quarter million if you can get one to us. And so Zerodium turns around and says, ah, well, we can resell that several times over. So let's bump the bounty for this to half a million dollars, knowing that, I mean, half a million dollars. That would be some significant incentive. It's not like, oh, yeah, we'll pay you 10 grand if you stumble over a browser escape or something. It's like, okay. So you could imagine that's going to have an effect.

In this acquisition raid, they bumped the payout for a high-quality solid and stable hypervisor exploit from what it had been, 200 grand, to $500,000. ZDNet in their coverage asked Zerodium's CEO what the deal was. He replied by email, saying: "Our new payout for hypervisors will last for a couple of months, and then we'll decide if we reduce it or keep it high, depending on the number of acquisitions we make." And that kind of gave me a chill. It's like, whoa, the number of acquisitions, suggesting not one, but oh, look, we got five new zero-days.

Anyway, Zerodium has previously held such acquisition raids for zero-days in iOS, in instant messaging apps, in the Tor browser, a.k.a. in the Firefox version of the Tor browser, in Linux, in Adobe Flash Player, in routers and USB thumb drives. And, okay, so the fact that they've done it for the Tor browser, you know, that sort of gives you a clue about who's on the other side buying them because they're not going to be bumping the bounty unless there's some reason to believe that they've got a buyer or buyers of these specific things.

So what's really going on here, believe it or not, is it's becoming a bidding war. And Zerodium was being outbid. Last summer, we talked about this at the time, Microsoft made the decision, and I salute them for this, to buy up its own flaws when they upped the ante for Hyper-V bugs to a quarter of a million dollars. So Microsoft has been and is officially on the record saying we will purchase a bug in Hyper-V for a quarter million dollars. Well, if that's the case, first of all, that outbids Zerodium and other exploit buyers who are only offering - only, only - $200,000. And whereas selling a powerful zero-day to Zerodium would leave me feeling a bit compromised, I would feel 100% great selling Microsoft one of their own bugs. That would be wonderful for a quarter million dollars. Right?

Leo: Yeah, it's a win-win. Everybody wins, including Microsoft Windows users.

Steve: Yes, yes. And you're just selling Microsoft their own bug. Here you go. I'll be happy to take a quarter million dollars for telling you about a problem I found in your stuff.

Leo: Right. That's ethical.

Steve: That's cool. That's 100% cool. But Zerodium's CEO told ZDNet: "Microsoft's bounty for Hyper-V exploits is very attractive for researchers. However," he said, "VMware is not paying anything to zero-day hunters. We have decided to fill this gap," as he put it.

Leo: Oh, they're such nice guys.

Steve: Isn't that nice? We're going to fill that gap because nobody's paying anything for those problems in VMware. He says: "...and we've been paying $200,000 for such exploits. And," he says, "we've acquired many of them so far." He said: "However, we've recently observed an increase in demand from customers, and we have decided to increase the bounty to $500,000 to outbid vendors and all existing buyers."

Leo: Wow.

Steve: So specifically saying to outbid Microsoft, to double what Microsoft has been offering, Microsoft being an existing buyer of problems in their own product.

So Zerodium claims that the company's hungry customers are government and law enforcement agencies, and these do seem like government-level purchase prices. So again, we don't know how big the market is on the dark side. But there obviously is one if they're willing to pay half a million dollars for a Hyper-V or VMware ESXi escape to the host. And the idea that we'll see how many we get, then we'll decide if we want to keep them coming at half a million dollars each, or drop it back down to be, you know, I mean, doesn't make any sense. I mean, if they were offering it for a quarter million, and Microsoft was offering it for a quarter million, I'm selling it to Microsoft because selling it to Zerodium gives me a creepy feeling. I mean, Zerodium gives me a creepy feeling.

But again, you know, I mentioned them last week because if we had somebody who decided to make a career out of finding problems, and they did find a problem that was significant, I couldn't fault a bug hunter for selling it to the highest bidder. It's not fair, really; but if it's your bread and butter, I mean, if it's not something you just stumble upon, again, half a million dollars is half a million dollars. So, wow. Anyway, we are now having a bidding war is the bottom line between companies, well, at least in the case of Microsoft, who is saying, yeah, we'd rather buy them ourselves than, I mean, look.

From Microsoft's standpoint, look at it. Zerodium was offering 200,000 until now. Microsoft said, okay, we'll see you and up you 50,000. So Microsoft was at a quarter million. Being willing to outbid Zerodium for flaws in their own software, feeling that that's better than allowing Zerodium to purchase them for 200,000 and resell them who knows how many times to who knows who. Because there's no reason to believe that these things aren't going offshore. So Microsoft, I think, did the right thing. Is Microsoft going to outbid Zerodium? I don't know. Anyway, interesting to see what a weird market has developed around flaws. And, boy, there's money in them there flaws.

Last week, as predicted and planned and as we talked about several months ago, the NSA did release Ghidra v9. That was last Wednesday during the RSA conference. This is, I'll remind our listeners, their free, powerful, and mature interactive multiplatform reverse engineering tool. The official site - you should bring this up on the screen, Leo. It's neat. It's that second link there. G-H-I-D-R-A hyphen S-R-E dot org.

Leo: Lisa talked to the NSA at RSA and actually met one of the developers of Ghidra. They were all excited about Ghidra at the RSA.

Steve: Yeah. After our mention of it a couple months ago, I received all sorts of terrific feedback about Ghidra's back story and proper pronunciation because I was clueless. I think I was saying "g-hydra" or something. Anyway, several people said, "Uh, Steve, it's

Ghidra." Anyway, there's sort of an IDA, if you stretch, in the middle, and that's for Interactive Dis-Assembler. So that's where this word came from.

Leo: Oh, oh.

Steve: Yeah.

Leo: I get it. But it's also Godzilla's buddy.

Steve: Yes, exactly. Anyway, Ghidra has been used internally at the NSA and other similarly closely aligned government agencies, like the CIA we talked about a couple months ago, for more than 10 years, during which time it's been evolving and developing continuously. It will doubtless prove to be extremely useful for anyone researching the operation and security of closed source software, and of course including for reverse engineering malware. The NSA explained that their general plan was to release Ghidra to enable security researchers to get up to speed and get used to working with it, get this, before applying for positions at the NSA.

Leo: It's a recruitment tool.

Steve: Uh-huh.

Leo: Wow.

Steve: Yeah, isn't that interesting? Or other government intelligence agencies with which the NSA has previously shared Ghidra privately.

So as we explained when we first noted this coming release, Ghidra is a free alternative to IDA Pro - IDA, I-D-A, Interactive Dis-Assembler - which is a similar reverse engineering tool that's only available under a very expensive multi-thousand dollar, I mean, like several thousand dollars, priced license. So by being offered for free, and soon to be open source, they're still in the process of getting it ready for open source release. There is a link. You can find a Ghidra at GitHub, but it points you over to the site that you went to and brought up on the screen. So by being offered for free, of course, most experts expect Ghidra to snap up a big portion of the reverse engineering tools and market share immediately, especially since the reviews have been almost entirely positive.

On GitHub the NSA has this to say. They wrote: "Ghidra Software Reverse Engineering Framework." And that's what the SRE is, Software Reverse Engineering. They said: "Ghidra is a software reverse engineering framework created and maintained by the National Security Agency Research Directorate. This framework includes a suite of fully featured high-end software analysis tools that enable users to analyze compiled code on a variety of platforms including Windows, macOS, and Linux. Capabilities include disassembly, reassembly, decompilation, graphing, and scripting, along with hundreds of other features. Ghidra supports a wide variety of process instruction" - they actually meant processor, they wrote process - "processor instruction sets." And my god, I mean, I saw the 6502 is there. So if you've got an Apple II or a Commodore 64...

Leo: Oh, finally I can disassemble Chopper Command.

Steve: Exactly, or Dig Dug or...

Leo: Super Breakout.

Steve: Lode Runner.

Leo: Load Runner, ooh, I would love to disassemble that, yeah.

Steve: Anyway, "...and executable formats and be run in both user-interactive and automated modes." So literally it will emulate the processor, and you can run the code in Ghidra. "Users may also develop their own Ghidra plug-in components and/or scripts using Java or Python." They wrote: "In support of NSA's cybersecurity mission, Ghidra was built to solve scaling and teaming problems on complex SRE [Software Reverse Engineering] efforts, and to provide a customizable and extensible SRE research platform. NSA has applied Ghidra SRE capabilities to a variety of problems" - oh, don't you know it - "that involve analyzing malicious code and generating deep insights for SRE analysts who seek a better understanding of potential vulnerabilities in networks and systems.

"This repository" - meaning at GitHub - "is a placeholder for the full open source release. Be assured efforts are underway to make the software available here. In the meantime, enjoy using Ghidra on your SRE efforts" - I mean, I am so tempted to play, but I can't - "developing your own scripts and plug-ins..."

Leo: Steve. SpinRite 6, Steve.

Steve: I know, I know, "...and perusing the over one million lines of Java and Sleigh code released within the initial public release. The release can be downloaded from our project home page. Please consider taking a look at our contributor guide to see how you can participate in this open source project when it becomes available. If you are interested in projects" - here it is. "If you are interested in projects like this and would like to develop this and other cybersecurity tools for NSA to help protect our nation and its allies, consider applying for a career with us."

So anyway, Leo, this is going to be big. I mean, I'm sure that IDA Pro has been cracked, and you can get cracks for keys and so forth on the 'Net. There's just no way that a tool with that kind of profile that is being sold for that kind of money hasn't been cracked by the crackers and available. But having an arguably superior tool, as I said a couple months ago, this announcement was not good news for the folks at Hex-Rays that have been arguably extorting the market because they had a captive market for years. That's over.

"Installing Ghidra is as simple as unpacking a zip archive. The only requirement is a version of the JDK [Java Development Kit] 11 or later, which is needed to run the app's GUI. The tool's official docs have the following to say about installation." Well, or quasi installation. They said: "Ghidra does not use a traditional installer. Instead, the Ghidra distribution file is simply extracted in place to the filesystem. This approach has advantages and disadvantages. On the upside, administrative privilege is not required to install Ghidra for personal use. Also, because installing Ghidra does not update any OS configurations such as the registry on Windows, removing Ghidra is as simple as deleting the Ghidra installation directory." So you just dump the files out of the zip, and you're ready to go.

They said: "Besides an installation guide, Ghidra's docs also come with classes and exercises for beginners, intermediates, and advanced levels that will help" - I mean, the NSA is really going for this. Although this was probably all developed for partners like other people at the CIA and other law enforcement, where they wanted to do reverse engineering, but they were like, huh? Okay, we unpacked it. Now what do we do? So the point is there's a lot of support here to help people get it going.

And they said: "This will help users get used to the tool's GUI, which is very different from any similar tools," i.e., IDA Pro. So this is not good news for the Hex-Rays folks, but it was inevitable. And, boy, I mean, this is a strong offering. And Leo, I think this is going to be significant. The idea where you have a tool like this, I mean, I really think this is enabling of, for example, this is the perfect follow-on to last week's subject of a career as a bug hunter. You cannot bug hunt unless you have a way of looking at code. And here it is, free for the download, a potent, world-class tool for allowing someone to start looking into code and developing an understanding of what it's doing and how it works. I just think this is very cool.

Leo: Now they just have to release a fuzzing tool, and we'll have the complete set.

Steve: Exactly.

Leo: We can do all the hacking, all the reverse engineering, bug hunting, yeah.

Steve: Wow. Wow. Interestingly, Firefox is adding something to their mainstream browser which the Tor version of Firefox has added a while ago. That's an anti-fingerprinting technique called "letterboxing." And as I was reading about this, I was thinking, really? So I suppose that I underestimate how determined online advertisers and profilers are to track us. And I guess I'm not that worried. But I certainly get it, that knowing all the places that a user goes can reveal a great deal of information about them.

So I suppose I'm glad that my browsing habits are likely to put anyone tracking me to sleep because I'm not very interesting from a "where do I go with my browser" standpoint. But I also do get it that the idea of unseen tracking and profiling can deeply offend the sensibilities of many users. They are, after all, not being asked and not being remunerated in any fashion for indirectly providing the history of their use of the Internet to unknown and unseen third parties who are apparently making money from the collection of the data of, to whatever degree that can be determined, who they are and where they go and what they do. So my underestimation of the determination of the trackers is revealed by the fact that Firefox thinks this is necessary.

Okay. So the Tor Project uses a version of Firefox, as I mentioned. The Tor folks have gone to, and there I can understand it, the extreme lengths to protect the privacy which their users very clearly want. We know that JavaScript running on a web page, and specifically in advertisements where JavaScript runs, is able to query for the size of the browser's display window. This is super useful for making responsive websites, which dynamically alter their appearance to deliver a good experience, regardless of viewport size.

This is the way state-of-the-art sites are able to reconfigure themselves. The script that loads the page or that loads on the page immediately looks to see where am I? Am I on a desktop? Am I on a mobile platform? What's the shape of the window? How big am I? And then builds a web page, essentially, on the fly in order to suit the environment that it's in. So that's script that needs to know about its viewport. So definitely important. But the instantaneous size of the browser's window, not surprisingly, can also be a signal that leaks identifying and de-anonymizing information to web page scripts.

And of course this is especially true when a user has their browser's window at some intermediate height. I actually sometimes do. If I've got a sufficiently large screen, I'm not running edge to edge or top to bottom. It may be sort of floating a little bit. Which means that JavaScript running on any frame for any reason on any tab of that browser will see a much more unique and arbitrary height and width. And in fact I think the script can also get the coordinate of the upper left corner, which would give it additional information.

Anyway, the point is that that could provide substantial disambiguating information to anyone who was tracking with serious intent. As we've discussed in the past from time to time, the mainstream Firefox browser has been backporting various enhancements for privacy, which were initially developed over on the Tor side, back into the mainstream Firefox. And this will happen again in two months, in May, with the release of number 67 of Firefox. It will inherit this so-called "letterboxing." Letterboxing deliberately masks the browser window's true dimensions by rounding the window width and height down to even multiples of 200 pixels in width and 100 pixels in height, respectively, during window resizing, which generates a much less unique window dimension for its users.

And of course to do this they're having to jump through some hoops. So during dragging, gray space will be added at the top, bottom, left, or right of the current page as needed to fill in the gaps. And if that sounds like something you don't want, and I'm certainly not as concerned as perhaps I should be, I don't really want this. The good news is that this new behavior will be a capability only and will be disabled by default. Once we get to release 67 in two months, you'll have to set privacy.resistfingerprinting to "true" in Firefox's about:config page in order to enable this letterboxing, which gives tracking scripts one less exact measure to track.

I wanted to share what Mozilla wrote on their Bugzilla page about this. They said: "Window dimensions are a big source of fingerprintable entropy on the web." And I don't doubt that a bit. "Maximized windows reveal available screen width and height, excluding toolbars; and full-screen windows reveal screen width and height. Non-maximized windows can allow a strong correlation between two tabs" - now, that's something that had not occurred to me before, but that's also true - "in the same window." Then they said: "While bug 1330882 takes care of new window creation, it would be ideal to protect windows continuously, even if users resize or maximize their window or enter full screen."

They wrote: "For Tor Browser we experimented with an approach to dynamically round the viewport," that is, the content rectangle dimensions, "to multiples of 200 by 100. In our prototype, when a user drags the corner of a window, the viewport 'snaps' to the largest contained multiple of 200 by 100, leaving a temporary margin of empty gray space in the window chrome. Then, when the user stops resizing, the margin of the window shrinks to nothing so that the outer chrome tightly encloses the viewport again."

So what you would see would be you'd be smoothly dragging the lower right corner of the window, or I guess any corner of the window, and the contents would be sort of holding back. Say that you were increasing the size. The contents would be holding back until it had the ability to grow by a multiple of 200 in width or 100 in height, and then it would snap to that next size. And in the interim the margin would get filled with gray. So again, this is where I'm kind of thinking, whoa, really? Are you guys that worried about this? Apparently. Then, when you let go, the outer chrome would snap back to tightly enclose the inner viewport so you wouldn't be left with this gray outstanding.

So anyway, that's what they're planning to do. And it'll be available to anyone who is serious about wanting to reduce the amount of fingerprinting available to anyone who is watching them very closely in Firefox. As I said, I'm not that concerned about it myself, but I wanted to bring it up. And I'm sure I'll mention it briefly in two months when 67 lands.

Leo: You saw the thing Mozilla also did. They're really cranking it out. The new Firefox Send?

Steve: No.

Leo: Oh, yeah, this is a cool feature. Private, encrypted file sharing through - actually, you don't even have to use Firefox. You go to send.firefox.com. If you're just somebody off the street, you can send up to a gigabyte. If you sign into your Firefox account, 2.5 gigabytes. It gives you an anonymous URL which you can click, HTTPS downloads. It's a pretty nice feature for people who need to send files. Now, I know you had a system that sent much larger files.

Steve: Filemail that I had discovered.

Leo: But this is right in the browser.

Steve: But still, 2.5 gigs is great.

Leo: Ain't bad, yeah. And I think most people who use Firefox have an account.

Steve: And I have a Firefox, I mean, I am a Firefox user, and I do have an account because I used it to synchronize my tabs between locations. That's interesting. So I upload a file to them, they give me a link which I then send to somebody else.

Leo: That's right.

Steve: Do you know how long the file stays available?

Leo: It's temporary. Let me see if it says. They do say it's temporary, with a link that automatically expires. Let me upload something and see how long. It's drag-and-drop. Oh, you can expire it after one download to 100 downloads, or one day to seven days. And you can password protect it, as well.

Steve: Nice. Filemail is seven days. Ah. And you have to pay Filemail in order to add password protection.

Leo: No, that's all for free.

Steve: I'm glad you brought that up, and I'm glad our listeners all heard you.

Leo: It's a great, I think a very useful service. And, you know, 2.5GB is a lot, especially since you can break it up.

Steve: Yeah.

Leo: Yeah. Steve?

Steve: So send.firefox.com is just - it's not a button in the browser. It's a site.

Leo: I would guess a service for Mozilla, yeah.

Steve: Yeah, exactly. And so, for example, for me, bing, I mean, 2.5GB, that's plenty. And I like the fact that I can encrypt with a password. They say: "Simple private file sharing. Firefox Send lets you share files with end-to-end encryption, and a link that automatically expires." Now, wait. End-to-end encryption. So that would...

Leo: So they can't see it; right?

Steve: Right. But I'm thinking that means the recipient has to be able to decrypt.

Leo: Oh, that's interesting.

Steve: So the link that you - yeah, the link you get must be to send.firefox.com.

Leo: It is.

Steve: So that you - okay. So that then that means - so you send someone that link. They click the link. It brings them to send.firefox.com, which then allows them to - that allows your browser to decrypt the blob that comes down. Anyway, they say "...encryption and a link that automatically expires so you can keep what you share private and make sure your stuff doesn't stay online forever."

Leo: Which is, of course, what you want.

Steve: Yeah. And so they're sort of selling the idea of expiration, which actually is a practical matter for them. They don't want to have all these 2GB blobs sitting around, stuck forever.

Leo: Right.

Steve: Which is, you know, it made sense for Filemail, too. But I do like the idea that you can password-protect something that is sensitive and that it's encrypting it in your browser and then decrypting it in the recipient's browser. So I'm sure there's some browser-side JavaScript, probably in WebAssembly, that's doing encryption on the fly. That's just really cool.

Leo: Yeah, they're really - this is, they say, part of their manifesto. This is what they're doing. I guess this has been around as a test, but they're making it official, even say they'll have an Android app.

Steve: Oh, wait. So you give it a password. That would mean that you'd have to send the recipient the link and the password, which then they put into their browser in order to perform the decryption.

Leo: Right, symmetric encryption.

Steve: Yeah. Nice.

Leo: So, yeah, pretty cool. That's a nice service. These guys, this is really - I don't know if it's a nonprofit. They make a lot of money in that Google search bar there. But they really are certainly altruistic in their efforts. I think it's really good.

Steve: I'm very pleased with what they're doing, yeah. It's still my browser of choice because it's got integrated tabs. And Google just seems to refuse to do side tabs for some unknown reason.

Leo: You're a tab guy.

Steve: Oh, my goodness, I couldn't survive without them. Oh, and by the way, in case anyone's listening - oh, shoot. What's it called? I should have prepared. The tab tool that I use uses CSS to customize the tabs. And as a consequence of a recent evolution of the Firefox browser, Tree Style Tab. And I saw somebody somewhere said that they weren't happy with it because their tabs were too large. And like me, I want my little - I want itty-bitty skinny tabs so that I can stack a whole bunch of them. And it turns out it is CSS style sheet customizable with lots of sample CSS styles already pre-canned. So you can do all kinds of things. So for what it's worth, I really like that free style or Tree Style Tab.

Okay. So Chrome and Windows 7(32) were suffering an in-the-wild zero-day. Week before last, on Wednesday, February 27th, and coming to light a week later, so last week, last Wednesday, security researcher Clement Lecigne, who is with Google's Threat Analysis Group, discovered and reported a high severity vulnerability in Chrome that could allow remote attackers to execute arbitrary code and take control of the computers that the end users were using. And Google has warned that this zero-day remote code execution vulnerability is being actively exploited in the wild to attack Chrome users.

The vulnerability was assigned a CVE of 2019-5786 and affects the web browsing software for all major operating systems, that is to say that the Chrome vulnerability is present in all major operating systems - Windows, macOS, and Linux. And until a majority of Chrome's users have been auto-updated, the Chrome security team is deliberately keeping all technical details to themselves. They've only stated that the trouble is a use-after-free vulnerability in the FileReader component of the Chrome browser, which leads to remote code execution attacks.

And we've talked about this kind of vulnerability before. A use-after-free refers to memory that was temporarily, often briefly allocated by the operating system when the browser had something it needed to do. You can imagine that FileReader would be that kind of thing. After that memory is no longer needed, it is released. It's freed, that is, the use-after-free. So it's freed back to the operating system. But due to a coding error in Chrome, essentially an accessible pointer to that memory persisted, and it could be used by a sufficiently clever attacker to execute code of the attacker's choosing on the user's PC.

And so Google security said: "Access to bug details and links may be kept restricted until a majority of users are updated with a fix." And as we know, that won't take long because Chrome is constantly updating itself. I always have, since I do run Chrome, it's got stuff running on my machine 24/7, essentially, checking to see if there's anything new and updating itself autonomously. So I'm sure this wouldn't take long. And this is, again, an example of, sad as it is, the fact that one way or another today's software that has an attack surface, and as we know nothing has a larger attack surface than a web browser, today's software that has an attack surface has to be able to update itself within short notice.

Anyway, they said: "We will also retain restrictions if the bug exists in a third-party library that other projects similarly depend upon, but haven't yet fixed." So they're going to look to see how pervasive this is and make sure that it's not - if it's not just constrained within their browser, they may not tell us for a while, despite the fact that Chrome has long since eradicated the problem. Anyway, so this FileReader API is a standard JavaScript API that's been designed to allow web applications to read the contents of files or raw data buffers stored on a user's computer using either "file" or "blob," as they're called, objects to specify the file or data to read.

That was Wednesday of last week. The next day, on Thursday, we further learned that the active in-the-wild attack was actually leveraging a pair of zero days, the one in Chrome being the first, and a previously unknown zero-day existing, as far as we know, only in Windows 7 (32), but not affecting Windows 10. And notice that it was observed on the 32-bit versions of Windows 7. We don't know for sure that it doesn't affect other versions. And this is what I was saying I also don't know because Microsoft has known of it for two weeks. Well, that's not a long time in Microsoft's world to prepare a patch, depending upon what it is, for a Patch Tuesday that was today.

So they only had apparently two weeks' notice because that's all Google Research was able to give them. But it was actively exploited in the wild, so it would have had priority with them. It's a local privilege escalation in the Windows win32k.sys kernel driver that can be used as a security sandbox escape. So of course that's why it's of use. If you have a browser vulnerability, you may still, despite the fact that you've got a vulnerability, you may be constrained by the browser's own sandbox that prevents you from doing anything.

So you also need a sandbox escape, thus the second vulnerability in 32-bit versions of Windows 7. The vulnerability is known as a "null pointer dereference." In Win32k there's a function, MN get pointer to item from index, under certain special circumstances that the bad guys were able to leverage in order to pull off an exploit. So Google said: "We strongly believe this vulnerability may only be exploitable on Windows 7 due to recent exploit mitigations added in newer versions of Windows." And they said: "To date we've only observed active exploitation against Windows 7 32-bit systems."

So then they finally said: "Pursuant to Google's vulnerability disclosure policy, when we discovered the vulnerability we reported it to Microsoft. Today, in compliance with our policy, we are publicly disclosing its existence because it is a serious vulnerability in Windows that we know was being actively exploited in targeted attacks." Ah, so targeted attacks, not spray, or not widespread.

And they said: "The unpatched Windows vulnerability can still be used to elevate privileges or combined with another browser vulnerability, that is, a theoretical browser vulnerability, to evade security sandboxes. Microsoft have told us they are working on a fix." So we don't know, again, if we made it into this Patch Tuesday. I should know next week. And hopefully, you know, Microsoft tends to be better, well, like way better than Apple, for example, in telling us what things they have fixed after they have been. So we may be able to know one way or the other.

I mentioned at the top of the show that Citrix was not having a good year. Last Wednesday Citrix learned - and this is a little obscure because they actually apparently learned earlier than this, but I'll explain. The news coverage said that Citrix learned from the FBI that their own corporate Citrix network had been infiltrated. When I read the blog posting by Stan Black, who's their Chief Security and Information Officer, now known as a CSIO, it didn't seem like that much of a deal. And again, I can understand him wanting to sort of downplay this.

His posting said: "On March 6, 2019, the FBI contacted Citrix to advise they had reason to believe that international cybercriminals gained access to the internal Citrix network. Citrix," he writes, "has taken action to contain this incident. We commenced a forensic investigation, engaged a leading cybersecurity firm to assist, took actions to secure our internal network, and continue to cooperate with the FBI.

"Citrix is moving as quickly as possible, with the understanding that these investigations are complex, dynamic, and require time to conduct properly. In investigations of cyber incidents, the details matter, and we are committed to communicating appropriately when we have what we believe is credible and actionable information. While our investigation is ongoing, based on what we know to date, it appears that the hackers may have accessed and downloaded business documents." Okay, now, that was the line that caught me up short when I learned what actually apparently happened. "It appears that the hackers may have accessed and downloaded business documents." Okay.

"The specific documents that may have been accessed, however, are currently unknown. At this time, there is no indication that the security of any Citrix product or service was compromised. While not confirming," he says, "the FBI has advised that the hackers likely used a tactic known as 'password spraying,' a technique that exploits weak passwords." And of course we've been talking about this recently. "Credential stuffing" is actually what it's called. "Once they gained a foothold with limited access, they worked to circumvent additional layers of security." Right. "Citrix deeply regrets the impact this incident may have on affected customers." And then I thought, what? What? Wait.

Leo: Hmm? But it didn't affect Citrix's products or services.

Steve: That's right. "Citrix is committed to updating customers with more information as the investigation proceeds, and to continuing to work with the relevant law enforcement authorities." So I was a bit curious about that second-to-last sentence, "Citrix deeply regrets the impact this incident may have on affected customers." Okay. Because of course as I said it appears that the hackers may have accessed and downloaded business documents. Anyway, I assumed that meant Citrix business documents; right? So that was Wednesday.

Leo: Ohhhhh.

Steve: Two days later, last Friday, NBC News posted the story with a somewhat different take than Citrix's own CSIO. NBC's headline and lead-in read: "Iranian-backed hackers stole data from major U.S. government contractor. The hackers are believed to have penetrated the software giant Citrix years ago and have remained inside the company's computer network ever since." Whoo. Ouch.

NBC said: "Iranian-backed hackers have stolen vast" - vast tracts of land, no - "...vast amounts of data from a major software company that handles sensitive computer projects for the White House communications agency, the U.S. military, the FBI, and many American corporations, a cybersecurity firm told NBC News. Citrix Systems, Inc., came under attack twice, once in December and again Monday, according to Resecurity, which notified the firm and law enforcement authorities.

"Employing brute-force attacks that guess passwords, the assault was carried out by the Iranian-linked hacking group known as Iridium, which was also behind recent cyberattacks against numerous government agencies, oil and gas companies and other targets, Charles Yoo, Resecurity's president, said." And here it comes. "The hackers extracted" - are you sitting down, Leo? - "at least 6TB of data and possibly up to 10TB..."

Leo: That's a big document.

Steve: "That's 10 [trillion, trillion, 10 trillion] terabytes in the assault on Citrix, Yoo said." That's Y-O-O. "The attackers gained access to Citrix through several compromised employee accounts," he said, "so it's a pretty deep intrusion, with multiple employee compromises and remote access to internal resources."

So between 6 to 10TB of something was exfiltrated from Citrix. This sort of sounds, though, as thought it might be their customers' business documents. And, yeah, Citrix would likely regret the impact that this incident would have on their customers. The security company, of course, in question was Resecurity. And in their own reporting of this - I've got a link to their blog posting in the show notes - they indicated that Citrix - and this is what's a little troubling - was first notified of the presence of an Iridium APT [Advanced Persistent Threat] last December.

Resecurity wrote: "Friday, December 28, 2018 at 10:25 AM. Resecurity has reached out to Citrix and shared early warning notification about targeted attack and data breach. Based on the timing and further dynamics, the attack was planned and organized specifically during Christmas period."

So what's odd is that Citrix's CSIO, who presumably would have been the recipient of Resecurity's note back on December 28th, stated in his blog that on March 26th, 2019 the FBI contacted Citrix to advise they has reason to believe that international cybercriminals gained access to the internal Citrix network. Yet Resecurity makes it very clear that they notified Citrix of this three months earlier. So one wonders whether Citrix just ignored the first report. But in any event, a multiyear targeted 6 to 10TB data exfiltration makes this a major breach of a major cybersecurity services company. Yikes.

Okay. And why you never want to loudly proclaim that "Our security cannot be hacked." Pen Test Partners' blog posting is titled: "Gone in six seconds? Exploiting car alarms." The two most popular aftermarket car alarm systems in the world open their owners to hacking. Who would have guessed?

Pen Test Partners' blog begins: "Key relay attacks against keyless entry vehicles are well known." And of course we've had, you and I, Leo, have covered key relay attacks extensively in previous years on this podcast. Their blog says: "Many third-party car alarm vendors market themselves as solutions to this." In other words, oh, key relay attacks are bad. You need to install this alarm system to be secure.

"We," they write, "have shown that fitting these alarms can make your vehicle even less secure." No surprise. "These alarms can expose you to hijack, maybe allow your engine to be stopped while driving, and it may even be possible to steal vehicles as a result."

Okay. So the car alarm systems are those by Pandora and Viper, and Viper is known as Clifford in the U.K. And I bought, shortly after getting the very first version of SpinRite up and going, I bought a Jeep Cherokee, and I equipped it with a Clifford alarm system. That was at the time the alarm that you wanted, digital encoded and all that. So I had a Clifford alarm.

Leo: Notice you don't hear car alarms go off much anymore?

Steve: No, thank goodness.

Leo: Used to hear them all the time. Remember?

Steve: Oh, they were so annoying, especially those ones that went through, like, 25 different...

Leo: [Mimicking car alarm] So annoying.

Steve: I know. Awful. Anyway, they have been found to be vulnerable, that is, these car alarm systems by Pandora and Viper have been found to be vulnerable to remote exploitation, enabling attackers to hijack the vehicles they're installed on and spy on their owners. The exploitable software flaws are found in the smartphone apps used to control - because of course we're going to give you an app for that - to control the alarm systems developed by Pandora and Viper. So get a load of this. The smartphone app has been downloaded more than three million times.

Pen Test Partners, who poked at the smartphone app, said that: "The vulnerabilities are relatively straightforward, insecure, direct object references in the API." They said that: "Simply by tampering with parameters, one can update the email address registered to the account without authentication, send a password reset to the now-modified attacker's email address, and take over the account." So in other words, it's apparently trivial to change the email address, then ask for a password reset, which then goes to the changed email address, which then allows you to take the account away from its owner. And then you have access.

Having access, playing with these parameters, you are now able to locate the car in real-time, disable its alarm, unlock it, enable or disable the vehicle immobilizer. In some cases you're able to kill the engine while driving. One of the two alarm brands allows drivers to be snooped on through a microphone in the car. And depending upon the alarm, it may be possible to steal the vehicles outright. And the flaws they observed in the car alarm APIs exposed huge amounts of personally identifiable information.

They also noted that it's not necessary even to purchase an alarm. That is, you don't have to own an alarm system. You're able to create a trial test and demo account online and then take over. From that demo account, it's possible to access any genuine account - it just sounds like it's horrible, like one of those put the email address as a parameter in the URL kind of things, I mean, just unconscionable security - and retrieve the user's details.

So in their blog - I'm going to quote some things just because it's amazing. Under "Killing car engines to order," they wrote, "This part is crazy. We discovered we could kill the engine on the Viper-equipped car" - that's the Clifford, known as Clifford in the U.K. - "whilst it was in motion. Promotional videos from Pandora indicate this is possible, too, though it doesn't appear to be working on our car." So they had a Pandora alarm-equipped car. "The intention is to halt a stolen vehicle. Except, using the account takeover vulnerability in the mobile app, one could kill the engine of any car fitted with these alarms.

"Audio snooping on drivers." They said: "Yes, really. The Pandora alarm has the ability to make SOS calls. A microphone is fitted in order to enable this. The microphone can be accessed and enabled remotely owing to the authorization flaw in the API. Therefore, all cars and drivers with the alarm fitted can be silently listened to. Millions of drivers snooped on."

And then under, they said: "CAN control." And of course that caught my attention because of course we know that the CAN bus we've talked about extensively on the podcast. They said: "CAN control. OMG! Both the Viper and Pandora have the ability to send custom CAN messages." They said: "This is where things get a bit scary. In recent years car alarms have the ability to interface directly with the CAN. This is necessary given the level of complexity," they write, "of modern vehicles. This also helps to reduce alarm wiring and installation time." Yeah, just plug it into the bus.

"Higher-end alarms can automatically detect the vehicle type they are being fitted to and customize their command set to the CAN messaging being used. This speeds up installation significantly." How convenient. "However," they write, "when the alarm doesn't recognize the vehicle, or it isn't automatically supported, the installer will need to program the alarm manually. As far as we can determine," they write, "alarm programming has to be done locally using a laptop app or, more interestingly, a mobile phone using Bluetooth. While there appear to be methods to program over the air from the API, they aren't documented fully, and we haven't been able to fully reverse engineer them yet. We are still working on this area, but each requires a different vehicle with an alarm fitted to prove it.

"But," they write, "Start/stop functionality is already enabled. After analyzing the firmware, manuals, and related changelogs, we found some scary functionality. It is vehicle specific, so we've been unable to test it all. If you own one of these vehicles with the relevant Pandora alarm, we would love to know if it works. Obviously, take great care; and we don't advise doing this on public roads. Mazda 6, Range Rover Sport, Kia Quoris, Toyota Fortuner" - I guess they meant Fortuner - "Mitsubishi Pajero, Toyota Prius 50, and the RAV4, these all appear to have undocumented functionality present in the alarm API to remotely adjust cruise control speed. Some workarounds for stop/start functionality also require a false brake pedal message to be sent, simulating the driver pressing the pedal and starting the vehicle."

So anyway, we covered the CAN bus and the power that it has in detail. As we know, and we talked about it at the time, there's an explicit and deliberate firewall which separates the infotainment side. There's essentially multiple CAN buses now with a careful firewall that allows controlled bridging of the two buses that separate the infotainment side from the critical vehicle operations side. It's clear that these extremely insecure alarm systems are being attached to the critical vehicle operations side, meaning on the dangerous side of the firewall, in order to give them the control they require.

So essentially, anyone adding this aftermarket alarm system is bypassing the CAN bus firewall and attaching a highly insecure and obviously very poorly designed system to the inner guts of their car. Fearing that the bad guys might already know of these vulnerabilities and be exploiting them in the wild, this Pen Test Partners outfit gave both companies a very short seven days to fix this most obvious security problem. Both Pandora and Viper, to their credit, responded and patched them immediately.

Pen Test Partners wrote: "Pandora's U.K. representative responded in about 48 hours and had their Moscow-based headquarters take action quickly. The insecure direct object references were fixed overnight, and we confirmed that the following morning. Viper responded even faster, but took a little longer to fix the vulnerability. That one is also confirmed fixed."

They also wrote: "It's important to note that we did not carry out a full test of the APIs. Doing that would have required further authorization which we don't have. We have no idea if there are other vulnerabilities in the API." In other words, they found the big horrific flaw in the alarm systems and notified the vendors. What they did not do was perform a full in-depth security audit. And, boy, that really is wanting. That should be done.

It's scary to imagine that, I mean, if this first obvious problem was found so easily, and these alarms are being directly plugged in over on the engine side of the CAN bus and can generate custom CAN messages, this is really ripe for exploit. And I saw the numbers, and I didn't put it in the show notes. Tens of millions of these alarm systems exist and have been installed. So a big target of opportunity.

Okay. So everybody who's listening to the podcast, I want you to raise your hand if you use the Alt+Tab Windows key combination.

Leo: I use it on Windows, Mac, and Linux.

Steve: Yay. You're a power user, Leo.

Leo: What?

Steve: Well, definitely some of us use it. But I'm sure it's a minority of Windows users. I use it often to quickly ping-pong between - typically between a pair of apps. Sometimes I'll use it to find an app that's deeper down in the MRU, the Most Recently Used stack. But I mostly use it to toggle between a pair which I'm interacting with. So Alt+Tab is my go-to or my "go back to the previous app" jump.

Leo: Right, right.

Steve: And so, for example, when I'm working on this week's podcast, I use an outliner called ThoughtManager Desktop, which is no longer on the market.

Leo: That's a vintage program.

Steve: Yes. It was on the Palm Pilot. It was an outliner for the Palm Pilot, and it had a Windows add-on component which was the desktop version. And you could export the things you did on the Palm Pilot to the outliner. But believe it or not, of all the outliners available for Windows, it is the simplest and cleanest, and it's where I do all of my - like all of my brainstorming and project organization is in ThoughtManager Desktop.

And so anyway, so I'll have it, and then I'll have Firefox. And as I'm pulling notes together and grabbing URLs and text from various web sources, I'm copying, and then Alt+Tab bounces me to ThoughtManager. Then Ctrl-V pastes it, and then Alt+Tab brings me back to the browser. And so I'm just - I'm bouncing back and forth all the time.

Anyway, my point here is that BleepingComputer had an intriguing piece about a new practice that Microsoft is exploring. Believe it or not, they've having and inviting five- to 10-minute conversations by appointment with Windows 10 users about their usage of specific Windows features. Lawrence Abrams, who covered this, wrote: "Microsoft has started to display notifications in the Windows 10 Action Center asking users to have a phone call with Microsoft developers and provide direct feedback about" - in this instance, there was a different one in February, he says, but in this instance - "about the ALT+TAB feature in Windows."

He says: "While using a Windows 10 Insider Build today, I was shown a Feedback Hub notification stating that: 'Microsoft wants to hear your opinions. To set up a phone call with Windows engineers, go to...'" - and then there's a link. Microsoft apparently has a domain www.aka.ms, which I thought was kind of cute, you know, also known as dot ms, and then /alttab. He says: "This link then redirects to a web page at ux.microsoft.com/?AltTab." He says: "It's not known if this is only being shown to Windows Insiders users at this time.

"When users visit this link they'll be shown a Microsoft User Research page stating that a Windows 10 product team is looking to 'understand our customer needs' and would like to have an anonymous five- to 10-minute phone call with the user. In this particular case, the phone call will be with Microsoft engineers to discuss how users use the ALT+TAB feature to switch between apps. Microsoft states they're performing these calls in order to get a better understanding of how a feature is being used while they're in development."

And so they said: "Your feedback is important to us. As we develop new software and services, it's critical that we get feedback from customers who use our products. Hearing from you early in the development cycle helps us make changes to our products and test them before release. Early customer interactions ensure we hit the mark with features. The time you spend with us today can improve our products for users around the world."

"According to the website," Lawrence says, "Windows engineers will be available on [yesterday] 3/11/2019 between 11:15 a.m. and 1:00 p.m. Pacific time, and [today] 3/12/2019 between 9:30 a.m. and 11:30 a.m. Pacific time, to schedule a call. The page goes on to say that users can expect a five-to 10-minute call, but it could last longer if there is more to discuss. They also state that the calls will not be recorded, are anonymous, and the content of the call will not be stored."

And then Lawrence wrote: "When researching this notification, I ran into a Reddit thread where a user received a similar notification over the weekend. Jen Gentleman, a Community Manager and Software Engineer at Microsoft, stated that these notifications are a new program being piloted that gives engineers the ability to talk to users in real time about features they're working on. So," he says, "this is not the first time that Microsoft has asked Windows 10 users to call engineers. In February, Microsoft also asked users to contact them regarding the Windows Update feature."

So anyway, I just thought that was fascinating that, okay, we're going to expose our cloistered and closeted Windows 10 engineers to actual end users. And I don't know if you get this notice if you don't use Alt+Tab? I mean, or...

Leo: What about Alt+F4? I think you should definitely talk to those people.

Steve: Oh, the close button, yes.

Leo: Yeah, the most obscure shortcut that people would use all the time if they only knew it.

Steve: Yup.

Leo: Wow. Geez.

Steve: But isn't that interesting? It's like, okay. Now, of course, I would say, just leave it alone. What, you know, why...

Leo: Stop messing with it, yeah.

Steve: Stop messing with it. But no.

Leo: No.

Steve: Yeah.

Leo: Wow.

Steve: So in other Win10 news, aside from Alt+Tab and who uses it, self-uninstalling updates is a new...

Leo: Oh, lord.

Steve: I know.

Leo: You shouldn't really need that, but okay.

Steve: I know. When you can't get it right, you can at least determine that Windows didn't start up and then back it out. Windows 10 will soon acquire the ability to autonomously uninstall updates that cause problems for it. Can you say "We are losing control of the boat?" According to a support document published yesterday, Windows 10 will begin automatically uninstalling automatically installed Windows updates that cause startup failures due to some incompatibility with something. You know, it's because it's just becoming too fragmented, and they're constantly changing it.

Updates will be automatically removed when Windows detects that it has recovered from a startup failure after all the other "automatic recovery attempts have been unsuccessful," whatever that means. And furthermore, to prevent an immediate repeat, because of course Windows will go, oh, I have an update missing, Windows will blacklist those from installing for the following 30 days.

So Microsoft wrote: "To ensure that your device can start up" - which is handy, you know.

Leo: Yeah. We would want it to start up, yeah. Good thinking, yeah.

Steve: "To ensure that your device can start up and continue running" - oh, yes, after starting up you would like it to continue running as expected.

Leo: Indeed. Indeed.

Steve: This is a new benefit, Leo, of Windows. It will continue running. "Windows will also prevent problematic updates" - those pesky problematic updates which we've had so much of lately - "from installing automatically for the next 30 days. This will give Microsoft and our partners the opportunity to investigate the failure and fix the issues. After 30 days, Windows will try again to install the updates." Wow.

Leo: This is actually really good because this does happen.

Steve: Yes. Yeah, no, it does happen, and it is good. But it's sad that it's necessary.

Leo: It's weird, yeah.

Steve: Which both you and I understand. "Users who still want to install them" - and who would that be? - "because they believe the removed Windows updates weren't the cause behind startup failures" - so that's some kind of power user - "can still do so by manually downloading them from the Windows Update Catalog or, in the case of device drivers, using the system's Device Manager to go get them."

So, yes, this feels like a useful and practical and unfortunately necessary CYA move for helping to deal with this past run of troubles Windows 10 has been experiencing after updates messed up the systems. And of course it's why I have to have my updates held off for a month because on this Windows 10 system I'm talking to you over, Leo, I don't want it to collapse on me on a podcast day. So anyway, there's that.

We finally have really good information about the Marriott Hotel's Starwood breach. The CEO, in some testimony in front of the Senate Committee on Homeland Security and Governmental Affairs, the Permanent Subcommittee on Investigations, Arne Sorenson, who's the CEO, apologized in his official statement and testimony to the company's customers, but also shot down some rumors that apparently were circulating that China was behind the attack.

So in his prepared statement, Sorenson said that Marriott learned something might be wrong for the first time on September 8th of 2018. So that was, what, late summer, early fall of last year, when they were contacted. They first learned when they were contacted by Accenture, the IT company who's responsible for managing the Starwood guest reservation database. And of course as we know and we talked about at the time, Marriott had acquired the Starwood Hotels properties two years previously in September of 2016. Work was underway to migrate Starwood's customers over to Marriott's own guest reservation system, but at the time the Starwood system was still separate from the rest of the Marriott network.

So on that September 8th, Accenture told Marriott's IT staff that a database monitoring system known as IBM Guardium had detected an anomaly on the Starwood guest reservation database the day before, on September 7th. Sorenson said - I guess it's Guardium. I have it twice here, so that must be correct: "The Guardium alert was triggered" - and I really was impressed by this - "by a query from an administrator's account to return the count of rows from a table in the database." It's like, well, that's very cool that it thought that was suspicious.

"Such queries," he said in his testimony, "are considered dangerous because the software that runs on top of a database doesn't usually need to make such a query." Again, I'm impressed. "This meant that a human operator was making this type of very specific query by hand." Sorenson continued: "As part of our investigation into the alert, we learned that the individual whose credentials were used had not made the query." At that point the Marriott staff realized they were dealing with a probable breach.

Leo: This is good. This is what's supposed to happen.

Steve: Exactly, although they didn't know if it was something big or just the beginning of a hack that could be very easily contained before the attackers accessed any user data. So it brought in a third-party forensic investigator, or third-party forensic investigators, on September 10th, so pretty much quickly, to help its IT staff look into a possible breach. Within a week, the forensic firm's investigation uncovered malware on the Starwood IT systems.

Sorenson testified that: "The investigators uncovered a Remote Access Trojan (RAT)," he writes in his testimony, "a form of malware that allows an attacker to covertly access, surveil, and even gain control over a computer." He says: "I was notified of the ongoing investigation that day, and our Board was notified the following day."

Significant forensic work was required to reveal the full scope of the attack. And despite the RAT's presence on the Starwood IT system, at that point there was still no evidence that unauthorized parties had actually accessed customer data from Starwood's guest reservation database. But the investigation continued. One month later, in October, the forensic firm discovered Mimikatz, a well-known penetration testing tool which searches a device's memory for usernames and passwords in RAM.

Leo: Wow.

Steve: Uh-huh. The tool was most likely used to help attackers acquire passwords for other Starwood systems to help them move laterally to other parts of the IT network. Still, however, investigators found no direct evidence that attackers had accessed customer data. They knew they'd had a bad infiltration, but they didn't have actual proof. A month after that, the hack turned from probably bad to definitely bad when, in November, investigators discovered that the hackers had been active on Starwood's IT network since July of 2014. This meant that a long-term persistent attack had been underway for more than two years without ever being detected. And this meant that the forensics firm had to dig through years of logs. Still, there was no actual evidence of attackers accessing customer data.

In mid-November, that evidence was finally found. Sorenson's statement reads: "On November 13th, our investigators discovered evidence that two compressed, encrypted files had been deleted from a device that they were examining. The files were encrypted, and the actual content was unknown. There was also evidence to suggest that those two files had potentially been removed from the Starwood network. Six days later, on November 19th, 2018, investigators were able to decrypt the files and found that one contained an export of a table from the Starwood Guest Reservation Database containing guest data, while the other contained an export of a table holding passport information."

The hotel chain then notified authorities and went public with its data breach disclosure on November 30th, revealing a breach that impacted around 500 million customers. According to Sorenson's testimony and an update on the Starwood breach notification website, the latest stats surrounding the Marriott breach, which have been updated several times as more was learned, are 383 million guest records were involved; 18.5 million encrypted passport numbers; 5.25 million unencrypted passport numbers, 663,000 from the U.S.; 9.1 million encrypted payment card numbers; 385,000 card numbers that were still valid at the time of the breach.

Sorenson stated that investigative efforts have yet to uncover evidence to suggest that attackers gained access to the encryption key used to encrypt the 9.1 million payment card numbers. But that also presumes that they were probably able to get the keys required to decrypt the other content. So it's likely that some Starwood employee somewhere fell victim to a phishing attack which allowed someone or something malicious to get into, well, something and someone malicious to get into their machine. That might have been a Remote Access Trojan that phoned home to report its success and then gave some remote attacker visibility into that machine.

From there the attacker likely perused the connected network and may have moved laterally to other machines. By downloading and installing and running Mimikatz in various machines, vestigial credentials could then have been recovered from RAM and used to further penetrate the organization's network, servers, and services. So now we know basically the full forensics breakdown of what happened with the Starwood properties.

Leo: That's actually fascinating. You realize how hard it is to catch this stuff.

Steve: Yes.

Leo: And now we know, we're pretty sure at least, the intelligence community seems to be pretty sure that these were Chinese state hackers on the Marriott one, that they were trying to kind of follow spies and Chinese nationals overseas in their reservations and so forth. Because it never went in the dark web and all of that.

Steve: Right.

Leo: So these were, one presumes, very, very skilled hackers.

Steve: Well, and delighted to, well, yes. And so the property was probably - it was a targeted attack, probably phishing email, which could have easily been designed to seem very authentic. All it takes is one employee somewhere...

Leo: That's all it takes, yeah.

Steve: ...to let it in, and then it gets a foothold and then begins, you know, then looks at the things that that user's machine's connected to, moves itself over into that machine, brings Mimikatz down to scan RAM, finds the login credentials of people who have previously logged into that machine. That allows the bad guys then to log into that machine. And as they say, that's all she wrote.

Leo: Yeah. Wow.

Steve: But, yeah, really, really cool to have a forensics readout of this.

Leo: Yeah. Fascinating.

Steve: And notice that we wouldn't have this if it weren't for Congress saying, you know...

Leo: Tell us, yeah.

Steve: Getting a public account from a CEO, how did this happen? Because no one wants to talk about this in this kind of details.

Leo: And yet, yeah, and I understand why, I mean, obscurity's security and all that.

Steve: Yeah, it's not in their PR interest.

Leo: Hackers know what they're doing, you know. And I'm sure they only left Mimikatz behind because, I don't know, why did they?

Steve: Didn't matter. Didn't matter at that point. They got what they needed. And, I mean, they were still there, Leo. They were there when this was discovered. So remember that it was somebody...

Leo: Well, they're going to stay there till they get kicked out; right?

Steve: Yeah. So again, it was somebody who, in real time, asked for how many rows in a table that tripped the IBM monitoring software that said, whoa, wait a minute, people don't ask for that. Or, I mean, sorry, automated software doesn't ask for that. Who wants to know? And that's what started this, I mean, that tripped the alarm that told them that somebody was there.

Leo: Wasn't that fascinating? Yeah.

Steve: So they were active at the time.

Leo: That's good security software, I think.

Steve: I'm impressed by that, yes. That's impressive. And I just got a kick out of this, so I wanted to add this to the show. Elaine sent back, in our communication that we have every week: "According to Merriam-Webster."

Leo: Oh, boy. I don't like it when emails start like that.

Steve: Thank you, Elaine. "While it's often maintained that the word 'doozy' derives from the Duesenberg in the name of the famed Duesenberg Motor Company, this is impossible on chronological grounds. 'Doozy' was first recorded, in the form D-O-Z-Y, in eastern Ohio in 1916, four years before the Duesenberg Motor Company began to manufacture passenger cars. The related adjective 'doozy,' D-O-O-Z-Y, meaning stylish or splendid, is attested considerably earlier, in 1903. So where did 'doozy' come from?" Entomologists? Etymologists.

Leo: Etymologists, yes.

Steve: "Etymologists believe that it's an altered form of the word 'daisy,' which was used especially in the late 1800s as a slang term for someone or something considered the best." Ooh, you're a daisy. It's a daisy.

Leo: Interesting. It's a daisy.

Steve: So it turned into "It's a doozy." Okay. I got a note from Jamie in Sydney, Australia, who said: "SpinRiting my iPod," he said, "working on one spot for hours and hours and hours." He said: "Steve, a bit of a story. There is a question in there somewhere, I promise." He says: "I've had a copy of SpinRite for ages. I purchased it many years ago, soon after having discovered the Security Now! podcast." He says: "At the time, I went back and started from number 1, so I've heard them all."

He wrote: "I've used SpinRite quite a few times around the house. I've even had to re-download it once or twice over the years as it is a very small exe that is very easy to lose." So, yes, you can misplace your SpinRite, and our system allows people who retain their purchase information to update their copy, to re-download it anytime they want.

So anyway, he said: "Recently I inherited an iPod." He says: "That's me justifying a short iPod Classic phase I went through where I picked up a few on eBay." He says: "I didn't realize at the time, but the hard drive had an issue. I mean, how are you to know, really? Oh, well, I figured I'd fix it with SpinRite. Didn't seem too much of an issue." Then he has a link to an article, "Running SpinRite 6 on macOS." He said: "Following this guide, I managed to get SpinRite running in VirtualBox, connecting to the iPod in Disk Mode. Very cool that we can do that at all."

He said: "I kicked off a Level 2 scan. As it progressed, it came up with an estimate of 33 minutes to complete. Things were chugging along, so I left it running and went on with some other stuff and sort of forgot about it. A couple hours later it's still running. SpinRite is fine, and I can navigate around the different UI screens, so the VM is still going, but we're still sitting on 14%. The time estimate has not moved."

And he says: "If I listen" - and here it is - "to the iPod, I can hear the drive in what seems like a loop. It's doing the same thing over and over again, chuck chunk chunk chunk squiggle berrrrrp squeak, over and over and over and over. I guess this is the drive trying and trying to read one troublesome spot."

"And so onto the question: When connecting to an iPod like this, or any USB drive, do all the special data recovery techniques you've mentioned over the years, coming at the spot from all angles, et cetera, still work? On the website you say 'It contains and deploys an extensive arsenal of data recovery techniques and technologies designed to pull a drive's data back from oblivion.' Do you still have the same low level control of the drive as you would with a direct IDE connection? Are you able to do everything to a drive connected in this way as you can to something directly connected to a physical PC via IDE/SATA?

"By the time you get this, and if it were to air on Security Now!" - oh, it is - "I may well have an answer of my own. I plan to just leave it running, at least overnight, maybe the weekend. Thanks for any answer, via Security Now! or not. Happy to share on Security Now! if you're running low on SpinRite stories. Long-time listener and enjoying every episode. Thanks. Jamie. P.S.: Could we bring back a few of the 'deep dive' style episodes where you go deep and tell us all about something?"

So anyway, to answer your question, Jamie, it is still possible to get better recovery if we have a direct connection. But many people, in fact many iPods have been recovered by SpinRite, and we've talked about that in the past. So anything is better than nothing. And you've got nothing. What you've got is clearly a drive that is very troubled.

And I remind people, and this is something of a value judgment, you know, some people, when SpinRite brings a drive back from the dead, they're like, oh, everything's great again. And it's like, uh, yeah, good. But you had a problem, and so you may have a problem again in the future. So it's not, I mean, it's great that you were able to get everything back. In some cases it makes sense to keep using the drive. But what I have long said is, in a competition between SpinRite and the drive, where SpinRite wants it to stay alive, and the drive is trying to die, ultimately the drive will win that competition.

Leo: You can't save a drive from itself. Everybody knows that.

Steve: If it is absolutely determined to stop being a drive and start being a doorstop, it will become a doorstop. So yes, SpinRite can help. It has helped lots of iPods in the past. Doing what you did, Jamie, is better than nothing. On my list, on my development plan, I'm first going to do the direct connect BIOS bypass AHCI connection for direct attachment of all of our IDE, AHCI, and SATA drives to motherboards.

But then my plan is - and I'm going to get that out immediately. And my plan is probably with .1, well, definitely with .1, probably do native Mac at the time because that would help a lot of people, too, who have drives in Macs. But then I absolutely want to do the serial interface. And so I plan to then do the same direct to the controller connection for probably I'll call it SpinRite 6.2 and release that again, do a series of consecutive releases. That's the plan. So that's where we stand. But yes, definitely. And as for the deep dive, I don't know how we were ever able to afford those, Leo, in terms of the podcast.

Leo: We can only do it when there's no news.

Steve: Yeah. And boy, that's just - give us a break, would you, world? And then I'd be happy to do a deep dive about something. But, boy, just no time.

Okay. So finally we're going to get to the topic of the show here at two hours in. It's called SPOILER. They have it in all caps. And it stands, well, I want to say it stands for, but it really doesn't, because it's "Speculative Load Hazards Boost Rowhammer and Cache Attacks." So Spoiler is not an abbreviation or an acronym or something. And I read somewhere that they just didn't bother trying. So the story is researchers from Worcester Polytechnic Institute in...

Leo: Worcester. Worcester.

Steve: Worcester?

Leo: Yeah, well, Worcester. But yeah, but they call it "Woosta." Worcester Polytechnic, yeah.

Steve: Oh, in Massachusetts. I get it.

Leo: It's like Worcestershire. It's "Woosta."

Steve: Okay. Worcester Polytechnic Institute in Worcester, Massachusetts.

Leo: That's it. Now you got it.

Steve: All right, and the University of Luebeck in Luebeck, Germany, named their work SPOILER. And I saw a comment somewhere that they were too tired afterward to even bother trying to reverse engineer that in order to make it into some clever abbreviation. So SPOILER is Speculative Load Hazards Boost Rowhammer and Cache Attacks. And so it's got buzzwords in it, but it doesn't stand for anything.

Okay. Now, their abstract, the abstract of their paper, is going to make everyone's eyes cross, I guarantee it. And it's going to stop the propellers of anyone whose propeller is spinning on their beanies. But that's also kind of the point I want to make with this. I want to show how the subtle and nuanced attacks such as Rowhammer and speculative execution evolve over time, and how far out into the weeds the academic security research community can and does go.

So for the simplified version, the abstract of their paper says: "Modern microarchitectures incorporate" - and I'm going to stop a few times to clarify. But "Modern microarchitectures incorporate optimization techniques such as speculative loads and store forwarding to improve the memory bottleneck." Okay, in other words, "speculative loads" means they're like reading ahead. They're like, as we know, fetching from DRAM, which is slow, in order to try to get content from memory that they believe they may be needing. So that's speculative.

"The processor executes the load speculatively before the stores, and forwards the data of a preceding store to the load, if there is a potential dependency." Okay, now, that means they've gone ahead and fetched something, but ahead of the logic storing something back into memory. So if where they're storing it back happens to be what they prefetched, then it's whoops, that would have changed the contents that they fetched if they had been waiting to fetch it, so that's a dependency. So that's known as store forwarding.

They said: "This enhances performance since the load does not have to wait for preceding stores to complete. However, the dependency prediction relies on partial address information, which may lead to false dependencies and stall hazards." They said: "In this work we are the first" - meaning the first researchers - "to show that the dependency resolution logic" - I mean, we're so far out in the weeds - "the dependency resolution logic that serves the speculative load can be exploited to gain information about the physical page mappings." Okay? Now that's of course the logical, the physical mapping. And that bears relevance because Rowhammer needs to know that the software is able to pound - it's trying to pound on physical memory, yet it's got to work through the page mappings, which has always been a problem for Rowhammer.

"Microarchitectural side-channel attacks such as Rowhammer and cache attacks rely on the reverse engineering of the virtual-to-physical address mapping. We propose the SPOILER attack, which exploits this leakage to speed up this reverse engineering by a factor of 256. Then we show how this can improve" one of the particular Rowhammer attacks known as "the Prime+Probe attack by a factor of 4096" which is really dramatic, I mean, it's not like it doubles it, it's 4,000 times faster "speed up of the eviction set search, even from sandboxed environments like JavaScript.

"Finally, we improve the Rowhammer attack by showing how SPOILER helps to conduct DRAM row conflicts deterministically with up to 100% chance, and by demonstrating a double-sided Rowhammer attack" - remember, that's where you hammer on both sides of your target, both rows on either side of your target row - "with normal user's privilege. The later is due to the possibility of detecting contiguous memory pages using SPOILER leakage." Whew.

Now, I actually had more here in the show notes, but I'm not going to dig into it because, I mean, it's - we're all exhausted at this point. Basically what has happened is this is yet another very clever attack on a different aspect of the Intel architecture speculation. We've talked endlessly through 2018 about branch prediction speculation and jump prediction speculation where there was explicit hardware present to prefetch and go down both paths of a branch and how it's possible to train the Intel speculation to expect to go down one path and then leak information when you do a context switch because you're using the same hardware as you were using as another process or VM, for example, was using.

Well, it turns out these guys recognized there was something else Intel had done to improve performance. And that's in this prefetching system where it turns out that Intel has a system which is responsible for this fetch ahead which is different than any of the speculation we've been talking about. And it turns out it can be tricked in a way which is new and unique. None of the existing speculation changes that we've made to microcode have any effect. They state that they don't expect to see for at least five years any change because it appeared in the very first core, the Intel core, the first-generation Intel core microarchitecture. It is fundamental to a much greater degree than these other speculative tweaks that Intel has made. It is like really deep in.

Microcode tweaks won't fix it. The only thing they could do would be to turn it off completely, and it would just crash performance if they did that. And what they did was they found this other aspect of crucial performance speculation which has been present in every core microarchitecture, and they figured out how to leverage it to create a far more potent Rowhammer attack by a factor of 4,000 than we've ever seen before. It is potent enough to work in browsers through JavaScript, which is a concern for even the Rowhammer mitigations where timers have been deliberately made fuzzy in order to thwart the attack. That fuzziness, the level of fuzziness needed was a function of how difficult the existing research had shown Rowhammer would be to work through an already uncertain environment like JavaScript, or to a lesser degree WebAssembly. These guys have got it working in JavaScript and in WebAssembly in a web browser. And so we are once again in trouble.

And Rowhammer, remember, is not something we ever really mitigated against. Well, there was mitigation in the browsers. There was talk about increasing the refresh speed so that the cells would tend less to be flipped by noise created by adjacent hammering on the rows. And then the possibility of next-generation DRAM being smarter about refreshing areas that were under read preferentially so this kind of problem - essentially make hammering attacks obsolete in the hardware.

But, you know, it takes years for that kind of mitigation to get into the channel and then into systems. And many systems are still using old-school inexpensive RAM that is very prone to Rowhammer-style bit flipping. And now every version of Intel core architecture, from core one, is subject to this kind of attack. So again, the famous words of Bruce Schneier echo: "Attacks never get worse. They only ever get better." And here's another example of that.

They did disclose their findings to the Intel Product Security Incident Response Team back at the beginning of December 2018 and probably ruined their holiday because there is just nothing they can do about this. This is really - oh, but no impact on ARM and AMD. This is explicitly an Intel memory system. It's called a "memory order buffer" is what they are exploiting. And so neither ARM nor AMD employ this particular form of speculation. But there are an awful lot of Intel core processors out there. And these guys don't expect them to be fixable. Yeah, yeah.

Leo: Well, fortunately, again, no exploits in the wild.

Steve: Spoiler, indeed. No.

Leo: Good work by the researchers once again.

Steve: Yeah. I would argue, you know, it's speculation we have never seen exploited. But we have seen Rowhammer exploited. Rowhammer works.

Leo: Right, right.

Steve: And so what this does is this leverages speculation to dramatically improve the efficiency and performance of Rowhammer. So this is, you know, it's like...

Leo: Interesting. They're inching closer.

Steve: ...where we combine problems and end up with something much worse than either one alone, yeah.

Leo: Very interesting. As always, the show is fascinating, and it's every week. And we've got, what is that, about 294 episodes left. So make sure you tell your friends. Security Now!, you can find a copy of it at Steve's website, GRC.com. He's also got transcripts from the dictionary-loving Elaine Farris. And you can get those at GRC.com. While you're there you can also pick up a copy of SpinRite, the world's best hard drive maintenance and recovery utility. What else can you do there? Oh, my gosh, there's a limitless amount of fun. Just explore around. Everything else is freely available, including SQRL and more.

If you want video, we've got it, along with audio, at our website, TWiT.tv/sn. And if you want to watch us live, we do it 1:30 Pacific, 4:30 Eastern. That is now 20:30 UTC because we sprang forward, 20:30 UTC at TWiT.tv/live. Audio and video is there. Join us in the chatroom if you do that because they're watching live, too, irc.twit.tv. And on-demand audio and video from TWiT.tv/sn. Or best thing to do, subscribe in your favorite podcast application. That way you'll get it automatically. Complete the set. Get all 999.

Steve, have a great night, and I'll see you next week on Security Now!.

Steve: I did want to mention we got a very nice response to my call for Android UX experts.

Leo: Oh, good. Oh, for SQRL.

Steve: Like seven or eight people have responded and said, hey, I can - oh, yeah, I said SpinRite, I meant SQRL, SQRL UX experts. So thank you very much, everybody. And they are online, and they're taking a look to see about enhancing the Android SQRL app. And I'm continuing to work on that, getting it ready for the world.

Leo: The world needs it.

Steve: Okay, my friend.

Leo: Have a great night.

Steve: Bye.