CPU.fail

Description: As expected after last week's Tuesday morning end-of-embargo on details of the next round of Intel processor information leakage problems, we will take a closer look at the new challenges they create and the impact of their remediation on system performance and stability. But before that we look at last Tuesday's patches from Microsoft, Adobe, and Apple. We examine a new big security problem for Cisco that even has stock analysts taking notice. We check in on the ongoing troubles with the cryptocurrency market, see what Johns Hopkins associate professor Matthew Green tweeted about the trouble with Google's Titan Bluetooth dongle, and deal with yet another monthly problem with Windows 10 updates. We touch on a bit of miscellany, then wrap up with a look at the new so-called Microarchitectural Data Sampling vulnerabilities.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-715.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-715-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve Gibson is here to explain all, to show us why access control lists may not always work as you expect, the story behind the failure of the Google Titan security key, and a patch for Windows XP? It's all coming up next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 715, recorded Tuesday, May 21st, 2019: CPU.fail.

It's time for Security Now! with Mr. Steve Gibson, the star of the show, the king of Internet security and privacy. There's no one better able - well, I kind of crowd you myself, but...

Steve Gibson: Yeah, thank you very much.

Leo: Yeah, at least I didn't put you on the Iron Throne.

Steve: I was just going to say, we didn't have any chance to talk about the "Game of Thrones" finale. And we don't want to do any spoilers for those who are...

Leo: We can't.

Steve: No, we can't say anything about it.

Leo: But you could tell me if you liked it.

Steve: Yeah. I mean, I just - I'm sort of sorry it ended. Like I could have just kind of kept...

Leo: I was so sad.

Steve: Yeah, I could have kept watching them run around in circles for, like, another 15 years. So...

Leo: I do think eventually people want to move on. Although, and I don't think it's a spoiler to say so, the way it ended left lots of room for sequels. And in fact I had heard that there will be. HBO is making some spinoffs and so forth. You could do both prequels...

Steve: No kidding. Good.

Leo: But, well, I'm not going to say anything about who or anything. But there were people at the end...

Steve: They can certainly reuse - they can reuse a lot of those costumes. They've got lots of spears. They've got lots of shields. They've got all kinds of stuff. It's like, hey.

Leo: Regardless of how you feel about the storytelling. It's, again, no spoiler to say a lot of people on the Internet - there's even a one million signature petition saying HBO should reshoot the whole entire thing, the last season. They blew it. I don't - I disagree with that. But I found it quite enjoyable. But like you, I was mostly sad that it was over. Somebody said, and I think this might be true, this may be one of the last times, if not the last time, the whole - a vast swath of people sits down on a Sunday night to watch a show because, A, fragmentation. There are going to be so many different - there are more and more places to watch stuff. And, B, a lot of these places dump the whole series all at once.

Steve: In binge mode, yes.

Leo: Yes. And so I wonder if there'll be an event like this again in our history. I don't know.

Steve: Well, the good news is Lorrie is someone like me who is able to watch something that's really good again. I've watched "Breaking Bad" now all the way through twice.

Leo: Oh, we're doing that. It's such a good show.

Steve: She fell in love with it. And so we're just, like, waiting for it to be long enough to watch it again. And so I'm so happy that I'm with someone. The woman I married years ago absolutely wouldn't even consider it. She says, "Well, we know what happens." I say, "But it's art. It's artwork. It's a process." "No." And it's like, "Okay, fine."

Leo: No, I'm with you. And Lisa and I in fact are watching "Breaking Bad" again. And what a great show. That was so well done. Man.

Steve: It was just - and it's the humor that was so much a part of it was they had fun.

Leo: Yeah. It was a great show. And the plotting and the acting and everything.

Steve: Anyway, no humor this week.

Leo: Oh, no. Oh, no.

Steve: No humor for you. Today's title is "CPU.fail," which actually is the website that was created to contain, not one, not zombie - I keep wanting to say "ZombieLand" - ZombieLoad. But rather four different new problems of which ZombieLoad was but one.

Leo: Oy.

Steve: So, yeah. And what's significant...

Leo: This is a gift that just keeps on giving. It's another one of those speculation errors; right?

Steve: Well, yes. Well, it's worse than that. Remember that, of the two, we had Meltdown, the Meltdown and then the Spectre problems.

Leo: Right.

Steve: Meltdown was easier to do and was a bigger problem. Spectre was always sort of more itself speculative and less obviously a big problem. Well, this is the logical extension of Meltdown into much bigger problems. I mean, basically I'm reminded that I'm stepping on my own dialogue from the end of this podcast. But Bruce Schneier gave us what I often quote him saying, "Attacks never get weaker, they only get stronger." The guys who did Meltdown basically never stopped working on this, and they made them practical. I mean, they reduced this to, yes, we can actually obtain a 128-bit AES key from, I mean, and they demonstrate it, from another process running in a shared hosting environment.

The good news is, first of all, once again the industry is taking it seriously. And also, this only really impacts, as has always been the case, environments where you have

inherently untrusted code sharing the same physical chassis. Which is not the typical listener. It's not the end-user mode. It's the cloud environment. And that's where the problem really is. So this has all been good for locking that down. But we have a lot of other fun stuff to talk about. We will end up talking about that in more detail.

Also, we now have some visibility into last Tuesday's patches from Microsoft, Adobe, and Apple. Microsoft and Adobe were biggies. Apple, you know, we never really get that much out of Apple. We know that they did respond to the microcode problem with patches. And we have a little bit of something from them. We also take a look at a big new problem that has just come to light from Cisco that's big enough that it's got stock analysts taking notice and saying, okay, like maybe this is going to have a financial impact on them.

Leo: Yikes.

Steve: Yeah. We check in on the ongoing troubles with the cryptocurrency market that we haven't dipped into recently. Also Matthew Green kind of lost it on Twitter over his reactions to Google's Titan Bluetooth dongle screw-up.

Leo: Oh, yeah, yeah.

Steve: But it's really fun. So we're going to follow his Twitter stream a little bit. Also we deal with yet another monthly problem that Microsoft just seems to be having now every month with their Windows 10 updates. We'll touch on a bit of miscellany, and then wrap up by talking about what is now being called "microarchitectural data sampling vulnerabilities," which is the umbrella under which all of these problems can sort of be aggregated - microarchitectural data sampling vulnerabilities, or as I also called it later on, "Speeding up is hard to do."

Here we have the Picture of the Week, Leo.

Leo: Yes.

Steve: And I just, okay, I love this picture. It's, you know, it's perfect. And we've done some things similar. I think I remember one where there was a path, a walking path, and someone had stuck a locking fence door on the path, and you could see the grass was worn on either side because people said, okay, this path is closed. I'm going to go around the door. I mean, it was just, like, so dumb. But anyway, that's our Picture of the Week is it shows this one-lane roadway that separates the outside, there's like public cars and things, like a public road, from the inside, with this dropdown arm, you know, the motorized arm that blocks the road. But there's nothing to prevent people from saying, okay, I'm in a hurry, I forgot my pass, I'm not supposed to go through here, whatever. And you can clearly see that it looks like nobody bothers to go through the arm. They just say "Forget this" and drive around.

Anyway, it's just wonderful. And so on one hand, okay, so here it is. I mean, this clearly exists somewhere in nature. You have to ask yourself, what were they thinking? Like what was, I mean, this cost some money, right, to like put this out there on this road. And to imagine that somebody is going to be inconvenienced, what are they are going to do? They're going to stop and go, "Oh, darn."

Leo: They just drive around it. But we've seen this before. We saw a gate, remember? You had a Picture of the Week that was a gate that people just went around; right?

Steve: Yes, yes. It's just, okay.

Leo: It just shows that, if you're going to build a fence, you've got to build a pretty long fence.

Steve: Yeah. Anyway, I've had this for a long time.

Leo: I love it.

Steve: And nothing else on the radar this week. So I thought, okay, it's time just to say, what were they thinking? Because this is not, I mean, this is - you could argue this is dumber than the password entry lock where the password is written above it. Because at least there, if you just changed your mind, you could take the password off and change the combo. Here it's like, okay, I mean, I guess they could strengthen this by putting fencing around.

I mean, basically nothing about this is defensible. I mean, like, there's no actual way, as you can see from the picture, to prevent somebody who's on the outside from coming onto the inside, or escaping, if that's what this is trying to - it's not clear what direction this is trying to control. Anyway, I just - this is just nuts. So I just - it's a great example of security not really working.

So Windows XP, Leo - and I said that correctly. Not Windows 8; not Windows 7. Windows XP was patched last week.

Leo: I know, I know.

Steve: For the second Tuesday of May.

Leo: Well, it may be more accurate to say a patch was offered because it wasn't automatic.

Steve: Correct, correct. So you know something is worrying Microsoft deeply when they're willing to reach back to Windows XP and produce, I mean, they don't even want to keep 7 current for like the Spectre and Meltdown stuff. We don't have, even now, complete updates for all versions of Windows 7 for that because there's a strong push, as we know. I've now twice had the warning come up that all security for Windows 7 is being discontinued, so we want to help you get your ducks in a row to move to Windows 10 before February of next year. And it's like, okay, well, my ducks are fine, thank you very much. We'll see what happens.

Anyway, so this is XP. Simon Pope, the director of incident response with the Microsoft Security Response Center, the MSRC, titled his announcement posting as a plea, with the

headline "Prevent a worm by updating Remote Desktop Services." He said: "Today Microsoft released fixes for a critical Remote Code Execution vulnerability" - it was given CVE-2019-0708 - "in Remote Desktop Services, formerly known as Terminal Services, that affects some older versions of Windows." And their announcement says "The Remote Desktop Protocol" - which of course we talk about often. "RDP itself," he says, "is not vulnerable." Okay, well, that's sort of a technicality.

He says: "This vulnerability is pre-authentication and requires no user interaction. In other words, the vulnerability is wormable, meaning that any future malware that exploits this vulnerability could propagate from vulnerable computer to vulnerable computer in a similar way as the WannaCry malware spread across the globe in 2017. While we have observed no exploitation of this vulnerability" - this is Microsoft saying - "it is highly likely that malicious actors will write an exploit for this vulnerability and incorporate it into their malware."

And I have to say that I love the fact that the world is finally sitting up and taking sober responsibility for these sorts of problems. We know that anyone can make a mistake, and that such mistakes can go unseen for years, as indeed this one did because it existed in Windows XP from day one, and sometimes even for decades. So it's not the fact of the mistake that matters, it's how those who are responsible for fixing it deal with that responsibility. So this is, I have to say, another bravo for Microsoft. On the other hand, they didn't really have much choice.

Simon continues his posting, saying: "Now that I have your attention, it is important that affected systems are patched as quickly as possible to prevent such a scenario from happening. In response, we are taking the unusual step of providing a security update for all customers to protect Windows platforms, including some out-of-support versions of Windows." He says: "Vulnerable in-support systems" - meaning those that are still receiving updates - "include Windows 7, Windows Server 2008 R2, and Windows Server 2008." But that means specifically Windows 8, 8.1, and 10 are not vulnerable to this. So this is older versions of Windows.

He said: "Downloads for in-support versions of Windows can be found in the Microsoft Security Update Guide. Customers who use an in-support version of Windows and have automatic updates enabled are automatically protected." So Leo, as you said, if you're in support and have auto updates, you'll be automatically protected. But unfortunately earlier versions, out-of-support systems, he says, "include Windows 2003 and Windows XP. If you are on an out-of-support version, the best way to address this vulnerability is to upgrade to the latest version of Windows." Uh-huh. On the other hand, if you're running XP, maybe you'll have, like, 2GB of RAM, and sorry.

Leo: Good luck.

Steve: Yeah. He says: "Even so, we are making fixes available for these out-of-support versions of Windows," and then there's a KB for anyone who's listening, KB4500705. So if you have XP. But it's worth saying, okay, well, we'll cover this in a little more detail about what this means in a second. But I was just going to say that, you know, you'd have to be exposed in some way. Your instance of Windows desktop remote terminal services, whatever you want to call it, Remote Desktop Protocol, would have to have an exposure to the public. So that's unlikely if you're behind a router. On the other hand, as we'll talk about in a second, there's a lot of those.

So they said: "Customers running Windows 8 and Windows 10 are not affected by this vulnerability, and it is no coincidence" - this is them stroking themselves - "that later

versions of Windows are unaffected. Microsoft invests heavily in strengthening the security of its products." I would argue it's probably a coincidence, but anyway.

Leo: Yeah, I find that hard to - well, we didn't fix it in any of these other versions, but we need to fix it on...

Steve: Yes, given the fact that what was it now, I don't have the number in front of me, 70-some - oh, there it is, 79 vulnerabilities were fixed in Windows 10 this month. Again. So they said: "Microsoft invests heavily in strengthening the security of its products, often through major architectural improvements that are not possible to back port to earlier versions of Windows. There is partial mitigation" - this is Microsoft - "on affected systems that have Network Level Authentication (NLA) enabled. The affected systems are mitigated against wormable malware or advanced malware threats that could exploit the vulnerability, as NLA requires authentication before the vulnerability can be triggered. However, affected systems are still vulnerable to Remote Code Execution exploitation if the attacker has valid credentials that can be used to successfully authenticate."

Okay. So what that means is that we've been talking recently about all of these problems with RDP. I've said to our listeners over and over, there is no safe way to have this port 3389 publicly exposed. You just can't. I'm a big fan of Remote Desktop. It's super convenient. Windows is inherently GUI by nature, so it wants GUI administration. Consequently, it's the way I remotely administer GRC's servers. But you will find no port 3389 or any other port listening on GRC's network that's got Remote Desktop Protocol exposed. There is no possible safe way to have it exposed.

So, okay. So RDP runs over a well-known port, 3389. So, not surprisingly, scanning the Internet's IPv4 address space for instances of that port accepting TCP connections is trivial. Now, get this. About 11 million, 11 million IPv4 addresses are currently accepting connections on port 3389. And because people have looked more closely after their eyes bugged out, somewhere around 4.1 million of those IP addresses respond as RDP servers today, 4.1 million exposed RDP servers. That's insane, but it's the case. And it certainly explains and warrants Microsoft's deep concern when we absolutely know that some probably significant percentage of those will be XP, Windows 7, Server 2003, Server 2008, or Server 2008 R2 machines.

So what we now know is that there is an exploit which will certainly be reverse engineered which allows all of those 4.1 million RDP-responding servers that are XP, 7, or any of those servers responding. This exploit allows them to be taken over remotely. And if they happen to have NLA enabled, which is not the norm, but if they did, then even those that we've been talking about whose credentials have been brute forced can now have remote code execution leveraged against them in order to allow the attackers to get more than they were able to before. So although on the other hand if you have a remote desktop protocol that gives the connector sufficient privileges, you've got effective remote code execution anyway because you can send anything through that connection that you want to and then run it at the other end.

So, okay. So just for the sake of our listeners, anyone who wishes, any individual or enterprise who wishes to make Remote Desktop Protocol available remotely must simply place it behind a VPN. That's all you have to do. Install an instance of OpenVPN server using certificate-based authentication. Set up an instance, one or more instances of OpenVPN clients with authorized client machines, giving them individual certificates signed by the server's private key. Now you have a robust means of managing a set of VPN clients who you can then selectively give access to remote desktop protocol through the VPN.

Again, unfortunately, I mean, it's almost unfortunate that there's even a password on Remote Desktop Protocol because, if there weren't one - well, I guess I don't mean that because people still would put it on the public Internet and go, oh, well, I guess we're not supposed to password-protect it. But the point is the password doesn't work, as we've seen, except on more recent versions of Windows. Now anybody can get to these things. So given that there are many millions of probably exploitable RDP servers that will probably not be updated, we're almost certainly going to see this. We'll be talking about this a month or two from now, the RDP worm that hit.

To their credit, again, Microsoft did what they could. They offered a patch to old systems that are probably online; but, as we have so often seen, probably not going to get patched, probably never going to get patched. And now they're going to get taken over as soon as some miscreants reverse engineer this problem, knowing first of all that there is a patch, looking at the patch, see what got changed. I mean, all the tools are in place. The skill set is in place. It's going to happen.

So we don't know that they will create a worm. In this day and age they'll probably just infect them with cryptominers and hope that they're running powerful servers that they can use for mining cryptocurrency, and they'll just suck all of the CPU cycles out of those. Or maybe they'll do a worm. The point was that Microsoft said it was wormable because it required absolutely no user interaction, just like WannaCry, where if it was able to get into a system, then it was able to do so autonomously, didn't require anything, to then be able to turn around and start scanning for more vulnerable systems.

Well, now we have Shodan, which is the way we know that there are 11 million open 3389 ports, and 4.1 of those respond to the remote desktop protocol. There's no question that a heavy percentage of those are going to be older machines that are not going to get patched. So we will likely be talking about the other shoe to drop on this one before long. And as I said...

Leo: [Sigh].

Steve: Yeah. Security is hard to do, Leo. Boy.

Leo: So this has probably been in this forever; right? I mean, this is not...

Steve: Yes, yes. It's been in the original Windows XP, when was that?

Leo: Oh, geez.

Steve: That was, what, 20 years ago.

Leo: Twenty years ago? Yeah.

Steve: Yeah.

Leo: So that's interesting, I mean, it doesn't...

Steve: It finally came to light. And so this is why, as we've been saying, it really is necessary - and we're there now, but we weren't there 20 years ago - for updates to automatically be installed, like for systems to reach out to somewhere and check to see whether there are updates available. And it's worth noting, I don't know if anybody has tried to set up XP. I do from time to time. In fact, I had to do it last week because somebody who was testing an update to the SQRL client for Windows, mine, GRC's that I've written, had a problem. I'd made some changes. I would have to look at my notes to remember what they were. But I had to set up a Windows XP.

Well, Windows XP no longer is able to even get updates because Microsoft changed the Windows Update format. And so you have to manually go get an update to Update, and update Update before it's able to update anything else. And so it's like, even the update won't update unless you first update it, which is sort of, you know, like wait a minute. Why?

Leo: Huh? Tell me that again?

Steve: Yeah. So anyway, Microsoft and Adobe both released their regularly scheduled patches last Tuesday, second Tuesday of May. Microsoft's was 79 vulnerabilities, 22 of them labeled critical. Of the 22 vulnerabilities, 18 affected their various scripting engines and browsers, and the remaining four were remote code execution in, as we've just described, Remote Desktop. There was also one in DHCP Server; one in the GDI+, the Graphics Device Interface extension; and also one in Word. And of course, as we'll be talking about later, Microsoft also released what they called "guidance" for the recently disclosed so-called Microarchitectural Data Sampling, now called MDS techniques, that we'll get to.

Adobe on its Patch Tuesday fixed one problem in Flash. It's nice that they stopped messing with it so they're not creating new problems, but they're still finding older ones. On the other hand, there were also problems in Acrobat and Reader and also Media Encoder. As I said, Flash only got one fix. Media Encoder had two fixes. Acrobat and Reader, 83 vulnerabilities in those. Because, again, if you just keep stirring the code, you're going to keep creating new problems. And that's what we keep seeing.

There were four remote code execution vulnerabilities. I did a little bit of digging, and I could not determine whether the Windows DHCP Server problem, which is a remote code vulnerability and is ranked critical, whether that affected it for its receiving packets from the public side, that is, on the WAN side, or on the LAN side. A DHCP server is, for example, may be standing in for DNS. And so it's telling all of the clients on the LAN side to refer to it for DNS. So then it's turning around and making DNS queries on behalf of its clients. That's sometimes the way things are configured. But it's oftentimes the case that DHCP Server will be making public queries.

What has come to light is that there is a way to provide a malformed reply packet to the Windows DHCP servers in order to cause a remote code execution vulnerability. As a consequence, for enterprise customers especially, this has jumped up to the top of the "must patch immediately." Here we are a week downstream. Hopefully IT people who are responsible already understood that this was important and fixed it because once again what we're seeing is the vulnerabilities are being reverse engineered and are being weaponized very quickly. The other problems, these other 22 vulnerabilities are remote code execution, privilege of elevation, and our regular cast of characters.

For the Zero Day Initiative blog post last Tuesday, Dustin Childs, who's Trend Micro's communications manager, wrote, speaking of these vulnerabilities: "They would first need to gain access to run code on a target system, but malware often uses elevations

like this one." So there was one that was not remote code execution, but it was an elevation of privilege, and we've been talking about this. They use elevations like this one to go from user to admin code execution. And again, there are no details provided for this. But they are being used in limited attacks in the wild. So again, this is one of those let's definitely get this patched. And with any luck, all of the people listening to this podcast have done so soon after, knowing that it was Patch Tuesday time.

I want to talk about an exploit that doesn't technically have a name, Leo. Unfortunately, the people who found this and have weaponized it and will be discussing it at length this summer at the Black Hat Conference 2019, decided to name it with three emojis.

Leo: Oh.

Steve: I know.

Leo: "An exploit has no name" would have been better.

Steve: Yeah. So much like the Artist Formerly Known as Prince...

Leo: Oh, impossible. What are the emojis?

Steve: So they are three angry cats.

Leo: And that's the name, Three Angry Cats.

Steve: So they're saying, if you must have some way to refer to it, they recommend Thrangry Cat.

Leo: Thrangry Cat.

Steve: Thrangry Cat.

Leo: That's a terrible name for an exploit.

Steve: It's awful. So let's take our second break, and then we will dig into something that - Thrangry Cat, yes, it's bad. And what's significant is, unfortunately, this is bad, enough that, as I said, financial institutions are worrying about Cisco's stock price because it affects hundreds of millions of deployed routers, switches, and firewalls. When you scroll through the list, I mean, it's like it's hundreds of individual devices that Cisco makes, deployed in hundreds of millions of publicly accessible locations, and it is remotely exploitable. So, yes, the cats are Thrangry.

Leo: Thrangry.

Steve: Thrangry.

Leo: Thrangry. I'm thrangry. Ah, Steve. All right, Steve. What's next? Thrangry.

Steve: So a new and very serious vulnerability present in hundreds of millions of Cisco routers, switches, and firewalls.

Leo: Geez.

Steve: Yeah. InvestorPlace.com noted: "Cisco earnings were great, but beware Thrangrycat. Cisco's latest security weakness is something to consider if looking to buy." So, I mean, it's really shaking things up. And I'm not that impressed with Cisco's handling of this. We'll get to that in a second. They go on to say that they have no idea what to make of Red Balloon's description. That's the group that found this and did some very seriously good engineering. I just wish Thrangrycat was, I mean, I guess I wouldn't mind if that was the official name. But, no, they've decided to name it as three emojis.

Okay. So the security firm is Red Balloon. They've identified a vulnerability, actually two, and in the first case have received a CVE of 2019-1649. And, yes, it's known - I have the picture of the three unhappy-looking cats in the show notes. Oh, and it doesn't look like they printed well. When I created the PDF, it just shows little blacked-out shadows on the show notes. So that's another reason why you don't want to [crosstalk].

Leo: It's not in mine. The computer you're using probably doesn't have the [crosstalk] cats.

Steve: Well, no, so you have the big three ones. But if you scroll down, I actually embedded the name throughout the text.

Leo: Yeah, yeah, it works, yeah. I think your computer doesn't have the requisite emojis.

Steve: Oh, you're right, it does show on yours.

Leo: It looks cute, in fact.

Steve: Okay. Well, are you on a Mac?

Leo: I am on a Mac.

Steve: Ah. So, Thrangrycat affects multiple Cisco products that support - oh, and Leo, if you want to, if you scroll down, there's a Cisco link. Look at the list of stuff in their disclosure. Oh, my goodness. It's just, I mean, the list of their hardware, it's everything they've been doing. Okay. So I want to back up. Thrangrycat affects multiple Cisco

products that support, okay, this is Cisco's own proprietary, they call it the "Trust Anchor module," TAm.

Leo: Is that secure boot or...

Steve: Yes. It's their version. As its name suggests, the Trust Anchor Module is a critical component of Cisco's hardware-based secure boot functionality which has been implemented in nearly all of Cisco's enterprise devices since 2013.

Leo: Well, that explains the length of this list.

Steve: I know.

Leo: All, basically, all.

Steve: I know.

Leo: Holy cow.

Steve: So they came up with this, and they said, oh, this is wonderful. Let's put this everywhere. Let's put this on everything. Secure boot. So it promises to ensure that the firmware, as we know, with secure boot, running on hardware platforms, is authentic and unmodified. Unfortunately, it turns out that's a promise it's unable to keep. In the words of its discoverers: "Red Balloon Security, Inc.," they wrote, "is disclosing two vulnerabilities affecting the products of Cisco Systems, Inc. The first, known as" - and there you have the three icons, the three cat heads, I'll just say Thrangry since that's what we have to do - "allows an attacker to fully bypass Cisco's Trust Anchor module via Field Programmable Gate Array (FPGA) bitstream manipulation.

"The second vulnerability is a remote command injection vulnerability against Cisco's IOS XE version 16 that allows remote code execution as root. By chaining the Thrangry and remote command injection vulnerabilities, an attacker can remotely and persistently bypass Cisco's secure boot mechanism and lock out all future software updates to the Trust Anchor module (TAm). Thrangry," they write, "is caused by a series of hardware design flaws within Cisco's Trust Anchor module.

"First commercially introduced in 2013, Cisco's Trust Anchor module is a proprietary hardware security module used in a wide range of Cisco products, including enterprise routers, switches, and firewalls. TAm is the root of trust that underpins all other Cisco security and trustworthy computing mechanisms in these devices." In other words, they put all their eggs in that basket, and then the eggs broke.

"Thrangry allows an attacker to make persistent modification to the Trust Anchor module via FPGA bitstream modification, thereby defeating the secure boot process and invalidating Cisco's chain of trust at its root. While the flaws are based in hardware, Thrangry can be exploited remotely." There again, the flaws are based in hardware. Thrangry can be exploited remotely without any need for physical access. "Since the flaws reside within the hardware design, it is unlikely that any software security patch will fully resolve the fundamental security vulnerability."

So we've not on this podcast talked about FPGA bitstream programming before, as we know. We have talked about FPGAs, Field Programmable Gate Arrays. An FPGA, field programmable, as its name suggests, is a means for creating soft hardware. It's a massive array of logical hardware elements - like AND gates, OR gates, selectors and such - that can be dynamically configured with RAM that is loaded into it. So the idea is that you have an outboard small EPROM or PROM which, at power up, produces a bitstream which this FPGA sucks into itself in order to turn it from just this massive generic unprogrammed blob into a piece of hardware.

And, I mean, the FPGAs these days have gotten unbelievably powerful. You can implement full processors in FPGAs that run at full processor speed. Because they're generic, they're not as efficient as if you actually did a masked programmed array, or actually designed the chip that you want. But you can do full-on real hardware this way. So the problem here is that something that Cisco did wrong allows this bitstream to be manipulated such that the hardware root of trust is corruptible. From their Q&A, answering the question how widespread is this, they explain: "This vulnerability affects Cisco products with an FPGA-based TAm. Cisco released their list of more than 100 product families containing this vulnerability." And as I said, Leo, it's a horrifying list. I mean, it's like everything that they've done for the last six years.

So they asked themselves the question: "What are the implications of demonstrating modification of the FPGA bitstream?" They write: "Our findings support the practical exploitation of FPGA-based devices via direct bitstream analysis and modification." So they reverse engineered the bitstream on one device that they got their hands on, took a look at it. And as is so often the case, you know, we've talked about how difficult it is to write code that interprets securely because the people writing the interpreter just - it's so difficult to take an adversarial posture. It takes an adversary to have an adversarial posture. Well, these guys have that. And so they looked at what Cisco did, and said aha, and realized that there were some problems here.

So they said: "Through our research we developed a series of techniques to reliably add, subtract, and alter FPGA behavior without any need to perform register-transfer level (RTL) reconstruction." Which is to say an RTL is one of the languages used for dealing with FPGAs. It's the way you describe this. So basically they didn't need to make a complete change to rewrite the behavior of the whole thing. They were able to edit some of that in order to cause a degradation of security, which probably is not that difficult to do. Any change that you make is going to do something that is non-optimal and is certainly not going to increase its security any. It has the reverse effect.

So they said: "By demonstrating successful FPGA modification on the Xilinx Spartan 6 LX45T" - which presumably is the FPGA that Cisco chose - "we find that our bitstream manipulation techniques present a range of potential applications for persistent FPGA implants, physical destruction of embedded systems, and attacks against FPGA-based systems, such as software-defined radios, advanced automotive driver assist modules, weapons guidance systems, and more."

Then they asked themselves in their Q&A: "Have these vulnerabilities been exploited in the wild?" They answer: "We are unaware of any use of this exploit in the wild, but the potential danger is severe." "What action can be taken?" They answer: "Please consult Cisco's official security advisory. We did not receive early access to Cisco's security patch, and will be analyzing the patches as they are made publicly available. Since Thrangry" - and, I mean, it is angry - "is fundamentally a hardware design flaw, we believe it will be very difficult..."

Leo: Oh, boy.

Steve: Yes, yes, yes. "It will be very difficult, if not impossible, to fully resolve this vulnerability via software patch."

Leo: Yikes.

Steve: So, yeah. Okay. So as I mentioned, I was unimpressed with Cisco's own vulnerability disclosure because that was in something of a panic. They downplayed the problem's severity. They wrote: "A vulnerability in the logic that handles access control to one of the hardware components in Cisco's proprietary secure boot implementation could allow an unauthenticated local" - it also works remotely - "attacker to write a modified firmware image to the component." Okay, that's technically true, but not only local. "This vulnerability affects multiple Cisco products that support hardware-based secure boot functionality." So, yes, right out of the mouth of an attorney. It's not incorrect; but, boy, it doesn't characterize the nature of the problem.

So Cisco continues: "The vulnerability is due to an improper check on the area of code that manages on-premise updates to a Field Programmable Gate Array part of the secure boot hardware implementation. An attacker with elevated privileges and access to the underlying operating system" - okay, both which these Thrangry people, the Red Balloon people, demonstrated - "that is running on the affected device could exploit this vulnerability by writing a modified firmware image to the FPGA." In other words, Cisco's basically confirmed what the Thrangry exploit is, which is the bitstream can be changed. The TAm module that anchors secure boot depends upon it not being changed. Therefore, secure boot can be bypassed.

Cisco continues: "A successful exploit could either cause the device to become unstable and require a hardware replacement, or allow tampering with the secure boot verification process, which under some circumstances may allow the attacker to install and boot a malicious software image." So since 2013 Cisco has had a secure boot trust anchor in hardware that now these people demonstrate across their devices, apparently all of them, based on Cisco's own list, can be compromised. There is a remote compromise available. So these guys are going to be demonstrating it in Vegas at Black Hat 2019. And they allege this cannot be fixed in software. So this has got to just be Cisco's worst nightmare. Wow.

Cryptocurrency hacks are still growing. We've not talked about, you know, we talked about bitcoin before it was a big deal. We talked about it, I'm sorry to say, back when it was possible for a Core i7 running overnight to generate a bitcoin, as mine did.

Leo: Or two, or 50.

Steve: Fifty.

Leo: Worth around $35,000 right now.

Steve: Yes. And then I reformatted the hard drive because...

Leo: Oh, you did? You killed the wallet?

Steve: I killed the wallet, yeah.

Leo: That was expensive.

Steve: Well, it was expensive because it peaked just shy of $20,000 per coin.

Leo: Yeah. That was really expensive.

Steve: Oh, boy, yeah.

Leo: That's, what is that, a million dollars?

Steve: Thank you, Leo. Yes.

Leo: What is that, like a million?

Steve: That was an expensive reformat operation. And, yes, I'm sure that the underlying data is gone because I then reinstalled Windows on top.

Leo: Did you have it in a wallet? Didn't you have it in a wallet, though?

Steve: Yeah, but I didn't take it seriously.

Leo: And you didn't keep the wallet ID or anything like that. Because you could, if you just had wallet.dat, you could get it all back.

Steve: No, that was on the hard drive. And so that's what got obliterated by a fresh install.

Leo: So kids, back up your wallet.dat. And by the way, if you password protect it, don't do as Leo did and forget the password.

Steve: Well, I'm sure back then I was using a password, one of a few that I'm almost sure I would have been able to use.

Leo: That's what I thought, yeah. And I did back up my wallet.dat. I have that. And I'm able to rebuild the wallet. If you just have that little file you can rebuild the wallet. But then you need to unlock it. I only have seven. You got 50, dude.

Steve: Yeah. Cryptocurrencies have enjoyed a recent resurgence. I don't know - I didn't look at it recently. I didn't look at it today. But they spent a long time wallowing down

around $4,000 for a bitcoin. And it's about doubled, last time I looked, around 8,000. So anyway, but where are we? We've talked a lot recently about Coinhive and about browser-based mining. We haven't checked in on the larger cryptocurrency world recently. There's a nice publication that I keep an eye on, the Hacker News. They have some good writers, and they pull pieces together sometimes. They wrote up an interesting summary snapshot with information gleaned from several different sources, including an outfit called CipherTrace that did a 2018 cryptocurrency report.

So with a bit of paraphrasing for length and a bit of editing for size, I wanted to share what the Hacker News wrote. They wrote: "Though once synonymous with underground networks and black hat hackers, bitcoin and other cryptocurrencies have gone mainstream over the past two years. In 2017, we saw the skyrocketing of bitcoin to an all-time high of close to $20,000, followed by a significant decline the following year.

"But beyond the ups and downs in the market for the world's largest cryptocurrency is a much more sinister story revolving around cyberattacks of the economy's newest asset class. In 2018 it's estimated that as much as $1.7 billion" - that's USD billion - "$1.7 billion worth of cryptocurrencies were swindled away from investors," and they said, parens, "(likely more) through a variety of means." And you and I have been talking about this, and we see this going on in the news all the time.

"Whether accomplished through hacking, phishing, or other forms of scamming, it's clear that the crypto industry is facing a serious dilemma with security. For a technological movement based on decentralization and the advantages it offers for security, the number of breaches occurring is startling." They wrote: "Cryptocurrencies offer users a way to send money without the need for a third party, yet the industry as a whole is dealing with more security vulnerabilities than centralized financial firms doing the same thing. During the same period, more traditional companies that transfer money and banks have seen nowhere near the same amount of issues with hackers. So what's the problem?

"While cryptocurrencies and blockchain technology are decentralized in nature, there are many aspects of the cryptosphere that aren't. The number one culprit in 2018" - and we know this from watching - "was cryptocurrency exchanges. Unlike the underlying technology behind currencies like bitcoin, Ether, and Litecoin, cryptocurrency exchanges are centralized and not yet regulated to the same extent as most financial firms. According to data from CipherTrace's 2018 Cryptocurrency Report, $950 million of the total $1.7 billion stolen were from exchanges and infrastructure services. Exchange services are a particular pain point for the industry because they're one of the easiest ways for users to get started with cryptocurrencies as some even handle fiat currency."

So anyway, I won't go on. I have more in the show notes if anyone is interested. But basically this tells us what we already know, and what we have seen, which is that we're seeing lots of startup me-too exchanges; and, if you build it, they will come. They collect a bunch of money, and whether it's an insider job - sometimes, you know, the people running the firms are accused of faking the loss or faking an exterior attack. Sometimes they probably are. As we know, security is difficult. And if these things are not well designed - and it's also the case that over time you generally mature your security. Hopefully it gets better over time.

So anything brand new, where there's lots of money involved, you're going to have people trying to poke it. And when it's a virtual currency, theft is electronic. So anyway, I just thought it was interesting to get a sense of scale. On one hand, sure, there's a lot of glamour behind this. On the other hand, boy, if you actually do have any substantial wealth in online coin, be very careful where you have it stored because, boy, it's like there's just been a long history of these problems.

Leo: Yeah. I mean, people would say, oh, Steve, you should just put it on Coinbase, and you'd be fine. Or Mt. Gox. If you guys would just put it into Mt. Gox, you'd be fine. Both of us had the good sense not to do that. I would feel a little bit worse if it was stolen than if we just lost it out of stupidity.

Steve: Well, Leo, I mean, remember, back then none of this existed.

Leo: Yeah, it was all made up, yeah, yeah, yeah, exactly.

Steve: I mean, to our credit, we did a podcast explaining what the blockchain was when no one had ever heard the term before.

Leo: True, true, true.

Steve: So, I mean, otherwise, I mean, I know that the drive was formatted because I went back and retraced my steps because for a while I was saying, oh, I think it might be on a computer back behind some boxes.

Leo: [Crosstalk] in the corner, yeah, yeah.

Steve: Yeah. And so finally I thought, okay, let's get serious about this, Gibson. And so I went back, and I absolutely verified, I know which machine it's on, I know which one it was on. I remembered running this thing overnight, waking up and going - and then I reported to our listeners, hey, remember when I said I was running a little miner? Well, turns out...

Leo: I remember it vividly.

Steve: ...the next day I had 50 bitcoin. It's like, woohoo.

Leo: Right.

Steve: Well, boohoo.

Leo: On the flipside, it didn't cost us anything to create them.

Steve: That's true. I do like to earn my money. But still, free money, ouch. Anyway, okay. So we have a little bit of comic relief here on a podcast that needs some. Matthew Green. Matthew Green, whose full name is Matthew Daniel Green, our oft-quoted cryptographer and security technologist. He's an associate professor of computer science at Johns Hopkins Information Security Institute. He got himself a bit worked up over the details of the recent flaw discovered in the configuration of Google's Titan Bluetooth security dongles.

So probably anybody who has one has already received email from Google. Google's very good about communication. Google is sending out replacements. If yours is marked with a T1 or T2 down in the bottom of the back of it, then it's vulnerable. I've got a link here, Leo, to Matthew's Twitter stream, but I also have it in the show notes.

So, first of all, Matthew quotes from the Bluetooth spec. He has a screenshot which I reproduced here. Reading from the Bluetooth spec. Get this, listeners: "The encryption key (K.sub.C) is derived from the current link key and may vary in single-byte increments from 1 byte" - what? Okay, the encryption key...

Leo: One byte? Let me see. There's 15 possible choices.

Steve: From 1 byte to 16 bytes in length, as set during a negotiation process that occurs between the master and slave devices. During this negotiation, a master device makes a key size suggestion for the slave.

Leo: Actually just one.

Steve: Yeah. How does that sound? Then you get to choose between any one of 256 keys.

Leo: Well, that should be enough for anybody.

Steve: Ah, who needs more than that? The spec says: "The initial key size suggested by the master is programmed into the controller by the manufacturer and is not always 16 bytes."

Leo: That's like a new kind of security; right? You don't know what it is.

Steve: That's right. "In production implementations, a 'minimum acceptable' key size parameter can be set to prevent a malicious user from driving the key size down to the minimum of 1 byte, which would make the link less secure." It actually says, "which would make the link less secure."

Leo: A lot less, yeah, yeah.

Steve: Oh, a 1-byte key? Really. So Matthew, now Matthew, after posting this screenshot from the spec, Matthew tweets: "Like, what kind of idiot protocol lets users negotiate a 'maximum key size' that can be as small as 1 byte."

Leo: Now, this is a Bluetooth spec, not a Google spec, we should point out.

Steve: Correct. This is Bluetooth. He says, parens: "(A default that, fortunately, should be higher in more recent versions.)" He continues, this is Matthew Green quoted on

Twitter: "Don't rely on Bluetooth security protocols for anything, ever. Just treat them like a particularly inefficient version of Base64 encoding."

Leo: I like this line. Keep going.

Steve: He says: "What would you possibly do with a 1-byte key? Is anyone at Bluetooth SIG even awake when these idiot decisions get made?" He says: "Hey, maybe we should call some experts and ask them if 1 byte is a good minimum size for an encryption key. Nah. We've got all the expertise we need right here. That's 256 whole keys." Then he tweeted: "Bluetooth" - and I didn't get this reference, but he explains it.

Leo: Oh, I get it, yeah. Yeah, yeah.

Steve: He says: "Bluetooth is the Michael Bay movie of encryption protocols." And I thought, huh? Then he explains: "It doesn't make any effing" - and he didn't say "effing," he used the whole word. He says: "It doesn't make any effing sense, and then it explodes."

Leo: That's a Michael Bay movie in a nutshell.

Steve: And then somebody tweeting as Tibor Jager, who is enjoying Matthew Green's tirade, adds: "I like the way they use the phrase 'less secure.'" To which Matthew tweets in reply: "What could it possibly be less secure than?"

Okay. So what's Matthew so worked up about? It seems that a team of security researchers at Microsoft discovered a serious vulnerability in the Bluetooth version of Google's Titan security keys - okay, now, remember everybody, these are the ones that everyone is using today, right, the T1 and T2 versions, right now, until last week - a serious vulnerability that cannot be patched with a software update. So Google has announced that it will replace all affected keys at no charge. Thank you, Google.

In their security advisory published Wednesday, Google said a "misconfiguration in the Titan Security Keys Bluetooth pairing protocols" could allow an attacker who is physically close to your security key, which is to say within 30 feet, to communicate - that's not bad, 30 feet - to communicate with it or the device to which your key is paired. And we know from Matthew that this misconfiguration allows the use of a 1-byte key.

As we covered at the time last summer, the Titan security key system consists of a USB dongle to provide hardware-based two-factor authentication for online accounts, which require the highest level of protection against phishing attacks. Google sells actually a pair - for the highest level you need two - sells for $50 in the Google Store a pair, which is a USB security key with NFC and a battery-powered micro-USB equipped for recharging Bluetooth NFC key which is used for multifactor.

Leo: That's really used on iOS because iOS doesn't support either USB or NFC. But everywhere else you can use the other secure key. But on iOS you have to use a Bluetooth key.

Steve: Right. And for something, it's one of their things that you have to have both, like the highest level of authentication.

Leo: You have to own both. If you want to use Google's, I forgot what they call it, but the high-end security - which I've done, I set up. And it doesn't have to be the Google pair. You can get a Bluetooth. But they want you to have both a Bluetooth security key and a hardware security key. But you don't need to use both. Either one will do.

Steve: So anyway, according to Google, the vulnerability only affects the Bluetooth Low Energy, the BLE version of the Titan security keys that have a T1 or T2, that's for Terminator 1 or Terminator 2, sign on - I don't know what it stands for - on the back. Other non-Bluetooth security keys, USB or NFC-supported versions are safe to use. So as Matthew says, he's not a fan, obviously. He would consider Bluetooth security, that phrase, to be an oxymoron.

Leo: They had to do it because of iOS. And that's less important now because Apple's opened up the NFC on iPhones. So it would really - it was for legacy support more than anything else. You could get away with just a YubiKey with the NFC now.

Steve: So the attack scenarios Google's Cloud Product Manager Christiaan Brand described in Google's blog post, he said: "When you're signing into an account on your device, you are normally asked to press the button on your BLE security key to activate it. An attacker in close physical proximity" - okay, but, you know, 30 feet - "at that moment in time can potentially connect their own device to your affected security key before your own device connects. In this set of circumstances, the attacker could sign into your account using their own device if the attacker somehow already obtained your username and password and could time these events exactly."

So, you know, not the end of the world, but it certainly defeats the multifactorness, which is why you have this whole thing in the first place. And then he continues: "Before you can use your security key, it must be paired to your device. Once paired, an attacker in close physical proximity to you could use their device to masquerade as your affected security key and connect to your device at the moment you are asked to press the button on your key. After that, they could attempt to change their device to appear as a Bluetooth keyboard or mouse and potentially take actions on your device."

So anyway, Microsoft, as I mentioned, initially discovered the vulnerability and reported it to Google quietly, as well as to Feitian, the company that makes these Titan keys for Google and also sells the same product under the trademark ePass, which is its own brand. They also made a coordinated disclosure about the vulnerability the same day as Google and offered the same sort of free replacement program for their own users. Anyway, so...

Leo: It's a Bluetooth LE issue, isn't it.

Steve: Yes.

Leo: This is part of the LE spec. LE's always made me a little nervous, to be honest.

Steve: And the spec has been improved subsequently. But again, why did it ever, why did it EVER say that you could negotiate down to a 1-byte key? I mean...

Leo: There must have been a reason, like some dumb hardware or something.

Steve: Yeah, exactly, or something really underpowered.

Leo: Remember keyboards. Keyboards. Remember they had - we mentioned this years ago, that a number of Bluetooth keyboards used, what was it, ROT13 or something for the encryption key.

Steve: Yeah. They weren't even Bluetooth, they were just low-energy radio.

Leo: Yeah, just wireless.

Steve: And so every time you hit a key, it incremented a counter that was XORed with the ASCII of the key. Oh, boy.

Leo: That's it, XOR, yeah.

Steve: I mean, that would be a beautiful, not even a grad project. It would be a...

Leo: Middle school. Middle school.

Steve: It would be a nice high school, something like a high school project.

Leo: Yeah, what's wrong with this? Yeah.

Steve: For a class, you know, crack this keyboard.

Leo: They're probably all still using them. I bet you there are plenty of them in there.

Steve: No doubt.

Leo: Yeah.

Steve: So we did have an iOS update. We're now at 12.3. Once again, I don't know what it is with me, maybe Apple just rolls it out slowly.

Leo: They do. They stagger them.

Steve: Yeah, because it wasn't, I mean, and this happened last week. But when I was pulling the show notes together last night, I thought, oh, really? So I went into general settings in my iPhone and said, "Got anything for me?" It said, "Oh, yeah, we got an update. Download and install." It's like, okay, fine.

Leo: Microsoft does this, too. Microsoft calls that being a "seeker." If you're a seeker of an update, they will give you the update. But they don't alert you to it until later. But you will eventually get an alert saying, hey, there's an update.

Steve: Oh, yeah. And I do on some devices that I don't really care about or use often. It'll be like, I'll look, and I'll see the little red, a little tag in the badge.

Leo: Yeah, yeah.

Steve: It's like, oh, I guess there's something I'm supposed to be doing here. So anyway, they fixed 42 CVE designations, 20 of which, not surprisingly or unusually, were in WebKit. So about half of them affected the public-facing side of iOS, which is the browser based. And as we know, Apple tends to be the most tight-lipped of all companies about their vulnerabilities.

There were a couple that jumped out. There was one, CVE-2019-8585, which affects CoreAudio. The suggestion that I've seen is that it might give malware a route to compromise an iOS device using a malicious movie file. If that's the case, it would be serious since it's probably not necessary for the target victim to do anything if, for example, they were just watching, I mean, if they were just receiving it through a text. Or maybe if it was displayed in a website it might be able to get into their system.

There's another one, 8593 was the AppleFileConduit, and 8605 in the kernel, either of which might allow someone to gain system privileges. So anyway, we don't know much more. Probably won't. I heard you talking, I thought it was interesting, Leo, on MacBreak Weekly about the prevalence of intrusions into iOS.

Leo: Oh, yeah, it's an interesting story.

Steve: How really at the state level there really is a lot of that going on.

Leo: Yeah, well, we just don't know because Apple's so locked down it's impossible to kind of look at an iPhone and see if it's infected.

Steve: Right.

Leo: There's no Process Explorer or anything like that.

Steve: Right. So I guess what you'd have to do is, what, just like do a complete wipe and reload from time to time, and then just be really careful about, I mean, the overwhelming, I think the best possible advice is not to just download everything that crosses your path if possible. Or if you do, you know, recognize that everything you download you are trusting to some level because it's very difficult for these systems to be kept secure.

I did want to mention, as I mentioned at the top of the show, once again we've got a problem that is occurring to some set of people after last Tuesday's Windows 10 Updates. Microsoft has acknowledged, and they're calling it a "known problem" - well, it's known now - that's causing some user systems to freeze. Leo, you may get people calling you on your show on the weekend. This occurs when users have created a restore point before applying the updates, and then for whatever reason subsequently decide they want to back out of the updates. Which of course should be fine since that's the whole point of having a checkpoint, having a restore point.

It turns out that after this past Tuesday's updates, when users attempt to do this, their system is bricked. They get a stop error. You cannot do anything more. There is a - I have it in the show notes - a link to, if this has affected anybody, or anybody who knows somebody who has been affected by this, there is a process you can go through. It's detailed and, I mean, it's not fun. You need to intercept the boot. I'm skipping over a bunch of this because I don't want to go into it in too much detail. Intercept the boot. Get into the Win10 restore environment, the rescue environment, and then jump through a bunch of hoops.

Microsoft explains, they said: "In this scenario, the system is not restored" - oh, so they said - imagine this. I mean, this is them speaking. "You cannot restore the system to a restore point after you install a Windows 10 update. Consider the following scenario: You install Windows 10 on a clean computer. You turn on system protection and then create a system restore point that is named R1. You install one or more Windows 10 updates. After the updates have finished installing, you restore the system to the R1 restore point.

"In this scenario, the system is not restored to the R1 restore point. Instead, the computer experiences a Stop error. You restart the computer, but the system cannot return to the Windows desktop." Under cause, they say: "This is a known issue in Windows 10. During the system restore process, Windows temporarily stages the restoration of files that are in use. It then saves the information in the registry. When the computer restarts, it completes the staged operation. In this situation, Windows restores the catalog files and stages the driver .sys files to be restored when the computer restarts. However, when the computer restarts, Windows loads the existing drivers before it restores the later versions of the drivers. Because the driver versions do not match the versions of the restored catalog files, the restart process stops."

So I have no idea what they just said except it bricks Windows. So for workaround they say: "To recover from the failed restart, after the failure occurs, you should be able to restart the computer and then enter the Windows Recovery Environment (WinRE). To do this, you may have to use a hardware restart switch, and you may have to restart two times." And then they go on. So for what it's worth, again, another month and another problem with Windows 10 updates. And boy, they really do seem to be having a problem with this.

Oh, also wanted to mention that yesterday, Monday, what would that be, May 20th, Microsoft Edge for macOS was officially released. There's a website, MicrosoftEdgeInsider, all one word, MicrosoftEdgeInsider.com. And now, under the platform, they're going to be doing a multiplatform release of Edge, not only for Windows 10 and eventually for Windows 7, but also for Mac. And so the Canary Channel version, which is the rawest, the least stable, is officially available for download. There's a

download button. The links to pre-release versions leaked a week or two ago, so some people were playing with it. So I'm not sure who would want to run Edge on macOS. But if there's some reason you have for doing that, it's now officially sanctioned.

Leo: I think a lot of people play with Chrome and would be interested in alternative, you know.

Steve: Yeah. Because, I mean, you've already got Chrome on macOS, so...

Leo: Right, right.

Steve: Yeah. So last [crosstalk].

Leo: [Crosstalk] want to use it is on Windows.

Steve: Yes. Yes, yes, yes. And there I think it makes lots of sense. Last week I mentioned that tomorrow I will be giving a SQRL presentation to the May OWASP Los Angeles monthly dinner meeting in the Los Angeles area. At the time there were 51 people signed up to attend. At this point we have 76. So I don't know if they're regular members who were planning to go and show up, or it looks like maybe we'll have a bunch of Security Now! listeners.

Leo: Oh, nice.

Steve: So if that's the case, I look forward to being able to say hi to some of our listeners who I never really have the opportunity to meet.

Leo: How nice.

Steve: So that'll be cool. Teams from the Graz University of Technology, whom we have spoken of before, KU Leuven, and Cyberus Technology have been at it again. And if we ever wanted, as I mentioned at the top of the show, a more perfect example of Bruce Schneier's sage observation that attacks never get weaker, they only ever get stronger, well, we have it here. What these guys have done, quietly - although they've known for a year, none of us have; they have been sitting, probably impatiently, on this - is a truly significant advancement to the practicality of what was the mostly theoretical attack on speculation and microarchitectural performance boosting that we began talking about right from the start of last year.

So the term "microarchitecture" is all throughout this, so let me just explain. The architecture is what the code that runs on the processor sees. So it's the registers. It's the stack, the various execution units. It's the view that the processor provides to the code. It was quite a while ago that that stopped being the only architecture in the chip, especially for the Intel chips, because the instructions are so complicated that, as they kept adding new features, the idea of designing this instruction set, implementing the instruction set as gates, just as simple logic, became impossible. Especially when they started wanting to get really fancy.

So what happened was they created a processor inside the processor, the so-called "microcode." So that runs, the microcode runs on the microarchitecture, which is this processor running at the clock speed that is given to the chip, 3GHz or whatever it is, and that processor essentially implements the instruction set that the outside world sees. So, and that's how, when Intel produces an update to the firmware of the chip, that's the microcode that implements this.

And so, for example, when they talk about microarchitectural data sampling vulnerabilities, the lid that's been torn off of the Intel architecture, unfortunately, is that there are all kinds of - because you have a processor in a processor. Because there's a whole 'nother dance going on separate from the one that's public. All of the attention up until now has been on the security of the public processor. That is, it's been scrutinized, and it's been looked at, and it's been validated as secure. Nothing leaks at that level. But as soon as we started looking past that, we got the Spectre and Meltdown problems at the beginning of last year.

So this is just, you know, this is so tasty for researchers. No one who's really into this has been able to put it down. And what came to light last Tuesday morning as this podcast was already set up and ready to go was essentially, I would say it's more than the next shoe to drop. It's an anvil dropped. Because what happened was the change from theoretical to absolutely proven. This is probably why Intel was really in a panic. Intel has known of this for a year. And so the good news is they've taken responsibility. I would, oh, my god, I would do anything to be a fly on the wall of, I mean, like, to know really what has been going on inside their meetings because, I mean, it has to be some serious discussion about how they got into this position and how these problems occurred.

But anyway, what we have now, after more than a year of trouble, is the so-called - it's sort of been generalized to "microarchitectural data sampling vulnerabilities." And to that original work, which first brought us Meltdown, we have on this site, CPU.fail, ZombieLoad, that we had fun just saying last week. We have something called RIDL, FALLOUT, and Store-to-Leak Forwarding. Each of those is a different exploitation of subtle features of Intel processors which have been produced over the last eight years, since around 2011, which as a consequence make it extremely difficult for it, the processor, to keep its secrets as it was always designed to.

That is, I mean, until the beginning of last year, we had these processors sitting on servers in cloud environments with VMs sharing a single core among any random set of parties, whether friendly to each other or not. That all changed at the beginning of last year because it turns out that at that time it was theoretically possible. A couple examples were shown. Mostly it was done like, oh, goodness, here's a problem, we need to fix this before it's too late.

Well, what we got just last Tuesday in the form of firmware updates for the microarchitectures, at some significant performance hit, I've seen some benchmarks that look like it takes maybe 10 to 12% off the top, which is a chunk of performance. Because unfortunately, as we've seen, in order to, you know, these problems are a result of the past of other threads of execution leaving trails, leaving hints in the microarchitecture which clever researchers are able to tease out over time. And again, these attacks have only gotten better.

Okay. So ZombieLoad, I'm not going to go into infinite detail because I just want to give our listeners a sense for this. ZombieLoad resurrects private browsing history and other sensitive data. It allows the leakage of information from other applications from the operating system across virtual machine boundaries in the cloud and from trusted execution environments. The RIDL attack allows the leakage of information across various security domains from different buffers such as line fill buffers and load ports.

Those are microarchitecture attributes that are not surfaced at the processor level, but they're all part of the plumbing that's going on behind the scenes.

RIDL demonstrates attacks on other applications, the operating system, virtual machines, and trusted execution environments. So there's a lot of overlap between these, but they are different types of attacks against different facets of the microarchitecture. The FALLOUT attack allows reading what the operating system recently wrote and figuring out the memory position of components of the operating system, thus strengthening other attacks. In other words, it helps to defeat KASLR, Kernel Address Space Layout Randomization.

And, finally, the Store-To-Leak Forwarding exploits CPU optimizations introduced by another microarchitectural component, the store buffer, to also break address randomization, monitor the operating system, or to leak data when combined with some aspects of Spectre. Spectre created these gadgets which were little pieces of code which were used. Store-To-Leak Forwarding is able to reuse some of these Spectre gadgets, essentially making the attacks stronger.

Microsoft, for their part, last Tuesday on May 14 wrote: "On May 14, 2019, Intel published information about a new subclass of speculative execution side channel vulnerabilities known as Microarchitectural Data Sampling." And of course now we know the data being sampled is somebody else's data, if you happen to be sharing an Intel processor with somebody else.

Microsoft said: "An attacker who successfully exploited these vulnerabilities may be able to read privileged data across trust boundaries. In shared resource environments such as exist in some cloud services configurations, these vulnerabilities could allow one virtual machine to improperly access information from another." So this sounds like a repeat of what we've been talking about all of last year. "In non-browsing scenarios on standalone systems, an attacker would need prior access to the system or an ability to run a specially crafted application on the target system to leverage these vulnerabilities."

What they didn't explicitly say is that code running in a browser is able to determine what the browser has been doing. So it's a potentially significant privacy violation just using a browser, at the browser level. So they allocated, or I should say the CVE system allocated four different CVEs: Microarchitecture Store Buffer Data Sampling, which was given the acronym MSBDS; Microarchitecture Fill Buffer Data Sampling, MFBDS; Microarchitecture Load Port Data Sampling, MLPDS; and Microarchitectural Data Sampling of Uncacheable Memory, MDSUM.

So over on the CPU.fail site they did a little Q&A to sort of help demystify this. They asked: "Am I affected by this bug?" The answer: "Most certainly yes." "Are these software bugs?" "No, these bugs are in the processor. Software can work around these bugs, which costs performance. Future processors will have integrated fixes." And that is no doubt true.

They ask: "Can I detect whether someone has exploited this leakage against me?" They say: "We have no data on this. The exploitation may not leave any traces in traditional log files." "Can my AV detect or block these attacks?" They said: "While possible in theory, this is unlikely in practice. These attacks are hard to distinguish from regular benign applications. However," they write, "your AV may detect malware which uses the attacks by comparing binaries after they become known." They ask: "Has this been used in the wild?" And they answer: "We don't know."

So they wrote a 16-page detailed paper that I'm not going to drag us through. And I've already pretty much explained what I had here, that I quoted from the abstract at the top of the paper. They conclude, though, saying: "With ZombieLoad, we showed" - and

this is in these 16 pages. "With ZombieLoad we showed a novel Meltdown-type attack targeting the processor's fill-buffer logic. ZombieLoad enables an attacker to leak recently loaded values used by the current or sibling logical CPU. We show that ZombieLoad allows leaking across user-space processes, CPU protection rings, virtual machines, and SGX enclaves. We demonstrated" - and they did this - "the immense attack potential."

And that's what has changed. I mean, the takeaway here is this went from being, well, okay, yeah, in theory this could happen. This research changed this to making these practical attacks. They said: "We demonstrate the immense attack potential by monitoring browser behavior, extracting AES encryption keys, establishing cross-VM covert channels, and recovering SGX sealing keys," Intel's secure enclave. "Finally, we conclude that disabling hyperthreading is the only possible workaround to mitigate ZombieLoad on current processors."

So for what it means to us, as I said, this is primarily a further collapse in interthread isolation. Interthread isolation only matters if, one way or another, a system has a malicious thread running on it. In the case of a user's machine, a malicious thread we would call malware. In that case, you've already got malware on your system, and so you're in trouble. This could be a way for malware to extract encryption keys from your system. Like, for example, if you were using BitLocker and depending upon BitLocker's encryption of your hard drive, it would be possible for malware to reach into the kernel to get the key that it would not otherwise have access to because it would be well protected, and then make off with it or send it somewhere, use it, do whatever it wants to. So but in general, once you've already got a malicious thread on your system, you're already in trouble.

The real problem, as I have mentioned, is in the cloud environment, where you might very well have a huge heterogeneous environment with a datacenter full of servers that are just mixing and matching and running anybody's code on any piece of hardware, moving VMs about without any concern. That's the way a lot of these systems are set up now. The problem is a bad guy gets in there on a system that has not been mitigated against this set of problems, and you've got trouble. I know under Linux with these patches applied and a state-of-the-art Linux, I've seen some benchmarks that look like it takes about, as I said, about 10 to 12% of performance right off the top. It looks like it's a larger hit on the later processors because the later processors more thoroughly take advantage of sophisticated microarchitecture in order to speed things up.

So as I said, my subtitle was "Speeding Up Is Hard to Do." Basically, I mean, I don't know what Intel is going to do. Probably what it means is that we're going to have to build into our operating system some explicit flushing or maybe an additional level of blinding between threads. Right now there is very little thread isolation between hyperthreads because hyperthreading is basically a second set of registers that you just - you do a very fast context switch from one bank of registers to the next. It was a clever, very inexpensive way of allowing a processor to continue running when it would otherwise have had its thread stall, waiting for something.

Intel said, hey, wait a minute, you know, with almost no additional logic we can create a lightweight context switch just by creating a second logical thread on the same hardware. Yes, they were right. But, boy, you don't want to have that other thread sharing the same core, which is what happens. You don't want to have it be malicious.

So already we have this notion at the architecture level of context switching. That's what we do when we push all of the threads on the stack and then load all the registers from a different thread that has been stopped. That's a context switch. What I think we're going to end up seeing is Intel having to, in order to save the fundamental design of their chips and implement a system which is secure, we're going to have to get something like a

context-switching mechanism at the microarchitecture level. It should have always had it. Nobody was looking. And nobody was this clever about and I guess really appreciated how much information could be obtained from little subtle variations in timing. At this point it really breaks the barrier, certainly between a pair of threads running on the same core. And of course, as we know now, since cores share caches, and we're able to probe cache contents, nothing is safe.

So at a cost of performance today, we're getting firmware updates. Microsoft has said they're only going to be doing it for Windows 10. So if you are running Windows 7 in an environment where you're at risk - and again, there's a bunch of caveats. In an environment where you're at risk, then it's worth updating your BIOS. I guess in general there's no reason not to update your BIOS if one is available from your manufacturer. But unless you are in an environment where you really are running threads that have a probability, a likelihood, a reasonable possibility of being malicious, if that's not your environment, all of this is interesting, and it's being fixed for you, but probably not a source of - it should not be a source of great concern.

Leo: Good.

Steve: Yes.

Leo: And I wonder, when they fix it with hardware mitigations, if they can do it without penalty.

Steve: I think we're going to get performance back, yes, because, I mean, this really did catch them off guard. And, I mean, as I've said before, I'm amazed they can fix this with firmware changes. I mean, that suggests that chips are way more programmable than I would have imagined. And so the fact that they can actually add what amounts to major features, basically there are feature registers that only the kernel has access to. Sometimes you only have, like, access at the BIOS level at boot time, and then it's all locked down. But they're adding bits that were never assigned to feature registers which are controlling significant aspects of the microarchitecture. I'm amazed that they can do that in microcode. I mean, it's like the whole thing really is deeply programmable, which surprises me.

Leo: Well, I mean, all they're doing is saying don't do speculative execution, in effect; right?

Steve: Yeah, but they never had a reason not to. So why let it be turned...

Leo: Right, why would they give - why is that switch there? Yeah, yeah.

Steve: Exactly, exactly.

Leo: Well, I mean, what it tells you is that a lot of the way processors work is not in the processor. It's in the microcode, which is programmable.

Steve: Right, right.

Leo: So, I mean, I'm being - this is a very stupid way of looking at it, but maybe they just did a branch around the speculative part, like there's a chunk of code that says, well, let's see if we can figure out ahead of time what he's going to do. Just jump over that stuff. Keep on going. Don't try. Don't attempt it. And the reason I ask if hardware fixes can still give you the performance is I imagine they're just going to eliminate speculative execution in the hardware, which would give you a hit unless they come up with something other than that.

Steve: Well, all they have - okay. Knowing what they know now, they could segment the history...

Leo: Right. We've talked about this before. Because the problem is a leak of information from one thread to another.

Steve: Yes, yes.

Leo: If you don't make that possible, if you hide it, then you don't have to worry about it.

Steve: Yes. If the context switching goes deeper into the architecture than just at the architecture level, if the context switching is pushed into the microarchitecture, then at a cost of significantly more complexity, you get performance, and you get interthread isolation, and that's what we need.

Leo: Right, right. Really interesting stuff. As always, Steve, thank you. I wanted to say hi to you from a guy named Aaron Miller. He's a college kid at the Buckeye Career Center in Philly.

Steve: Cool.

Leo: Big listener of the show. And he and his buddies just won second place in the national BPA Awards. BPA is Business Professionals of America. And they have a network design competition. And there they are with their - that's Aaron on the far right.

Steve: Oh, very, very cool.

Leo: So he says, "I owe it to Steve and Security Now!." But the other thing which I loved is his network design included Canaries.

Steve: Aha, very nice.

Leo: Not only does he listen to you, he listens to our ads. So Aaron wanted to say hi. He and his family, David and Sharon and Aaron were all visiting. Philadelphia, Ohio, by the way. If you wonder why there's a Buckeye Career Center in Philadelphia, PA, it's Philadelphia, Ohio - New Philadelphia, Ohio. Want to get that right.

Thank you, Steve. We see people all the time who say, "Say hi to Steve. Tell Steve I owe it all to Steve." A lot of people who are in IT say, "Yeah, I listen to Steve." In fact, I met a guy - you'll love this - on Sunday. No names. He's in the Air Force. He is at Stanford right now studying security. He is going back to work in North Carolina at Fort Bragg, where he'll be responsible for securing our nation's infrastructure against election fraud and other influence from other nations.

Steve: Nice.

Leo: And he says, "And I listen to Steve religiously." I said, "Wait a minute, you're operating at this high a level?" And he said, "Yeah, and I still learn a ton of stuff from Steve." So that's pretty high praise, Steve.

Steve: Neat. Very nice. Very cool.

Leo: Hey, we are done with Security Now!. You can go to the website, Steve's website, GRC.com, and get yourself a copy. He has regular 64Kb MP3s, but also 16Kb MP3s. They don't sound perfect, but they have the advantage of being teensy-weensy. But the smallest version of the show you can get from him is the handwritten, human-written transcript from Elaine Farris, who writes a very nice transcript of the whole show. So you can read that. Best solution is to read while you listen, and use your underliner and notes. And then maybe you can win second place in the BPA Nationals.

Steve also has lots of other things at GRC.com, his website, including of course his bread and butter, SpinRite, the world's best hard drive maintenance and recovery utility; and lots of information about other things, including SQRL, soon to be a major motion picture. Or something. He's been writing it for years. We are at TWiT.tv/sn. We have video there, and of course the best thing to do is subscribe in your favorite podcast program. That way you'll get a copy of it the minute it's available every Tuesday afternoon.

Quick reminder, we record Tuesdays, 1:30 Pacific, 4:30 Eastern, 20:30 UTC. You can tune in and listen or watch live at TWiT.tv/live. If you do that, hang in the chatroom, nice people who are all listening at the same time so you can kind of have a conversation about what we're talking about amongst yourselves. I'm in there, too: irc.twit.tv. Steve, thanks so much, and we'll see you next time on Security Now!.

Steve: Thank you, my friend. Till next week.